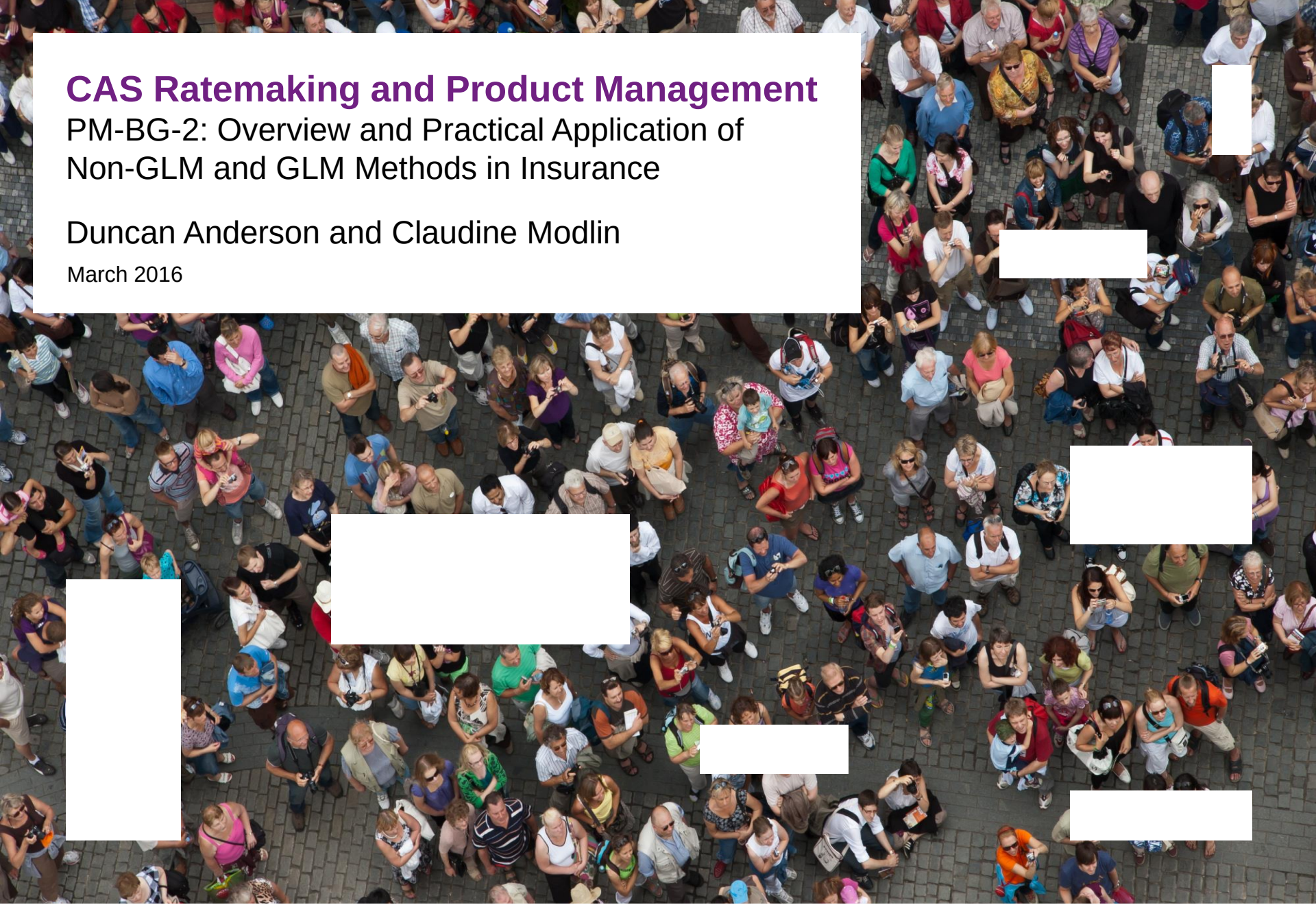
CAS Ratemaking and Product Management

PM-BG-2: Overview and Practical Application of Non-GLM and GLM Methods in Insurance

Duncan Anderson and Claudine Modlin

March 2016



Willis Towers Watson

Agenda

- The influence of Big Data on insurance
- Overview of analytical approaches
- Using & evaluating methods in practice
- Conclusions

The influence of Big Data on insurance

Mean Expected ROI on Big Data by Industry – **Insurance - 38.9%**

Tata Consulting:
'The Emerging Big Returns on Big Data'

Harvard Business Review,
October 2012

When companies inject data and analytics deep into their operations, they can deliver productivity and profit gains that are **5 to 6 percent higher** than those of the competition.

Dominic Barton and David Court

State Street
2013 Report ***Data and Analytics in the Insurance Industry***

" **Data leaders will dominate the insurance industry**. Nearly two-thirds of insurance executives agree that data and analytics capabilities will be among their most important competitive advantages in the future "

How to Build Analytics into the Insurance Value Chain

"A leading Insurer utilised customer behavioural analytics to **increase profits by more than 2% of GWP"**

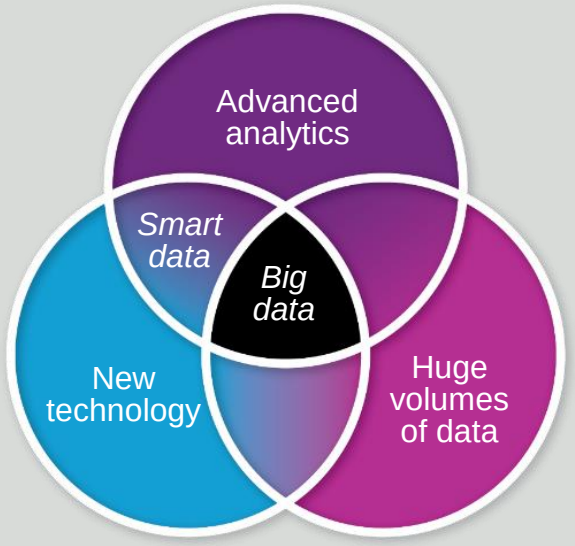
Asked participants whether their analytics initiatives had improved decision making in the business. The answer for the clear majority **– 80% was indeed yes.**

Tata Consulting:
'The Emerging Big Returns on Big Data'

23% of the insurance companies surveyed in the US **sell their digital data to third parties and finish well above other industries in terms of revenue generated**

Tata Consulting:
'The Emerging Big Returns on Big Data'

How is Big Data affecting insurance?



Overview of analytical approaches

Supervised vs. Unsupervised Learning

Supervised learning

- Requires factors (x_i) and response y
- Learns a function f : X → Y which is used to make predictions, eg:
  - GLMs, Penalized Regression, Decision Trees, Support Vector Machines, Neural Networks, k-Nearest Neighbors, Random Forests, Gradient Boosting Machines
- Regression models for numerical responses, and classification models for categorical responses

Useful	▪ Trying to predict something
Not Useful	▪ Accuracy and interpretability vary by method

Unsupervised learning

- Requires only factors (x_i) to infer the underlying structure of X, e.g.,:
  - K-Means Clustering, Dimensionality Reduction (eg Principal Component Analysis, Stochastic Neighbor Embedding)

Useful	▪ Exploratory analysis to understand correlations, patterns, variable importance ▪ Feature selection (particularly with large number of highly correlated variables) ▪ Saving memory/computation by reducing noise in high-dimensional data ▪ When responses can be harder/more expensive to obtain
Not Useful	▪ Trying to predict something (more a pre-processing step)

Generalized Linear Models

Model form

$$f(X) = g^{-1}(X\beta)$$

- g – link function
- X – design matrix
- β – fitted parameters
- Distribution specified by the error structure, but must be from exponential family

Model fit

- β calculated to minimize a loss function $L(\beta|X, y)$ (such as negative log-likelihood), i,e,. minimize:

$$M(\beta) = L(\beta|X, y)$$

- Most of the work is in specifying the design matrix, X

Useful	▪ Need a parametric solution that is interpretable
Not Useful	▪ Data exhibits high degree of non-linearity (with effort can capture non-linear effects in linear framework) ▪ Requires care when predictors have high dimensionality (e.g., postal code)

Penalized Regression (General)

Model form

- Same model form as a GLM, ie:

$$f(X) = g^{-1}(X\beta)$$

Model fit

- Differs to GLMs in the way that β is estimated
- An additional term is added to the objective function to restrict the magnitude of coefficients that may lead to overfitting:

$$M(\beta) = L(\beta|X, y) + \lambda \times \text{Penalty on } \beta$$

- For any value of λ, the solution for the penalized regression is obtained by finding the parameter estimates that minimize the objective function plus the penalty term
  - λ = 0 gives same answer as GLM (sensitive to noise in the data)
  - λ = *infinity* shrinks all parameters to 0 (biased toward the intercept)
  - λ is chosen such that the error on hold-out data is minimized

Useful	Avoid overfitting, generates insights into variable selection
Not Useful	(still a linear model)

Penalized Regression (specific types)

Ridge

- Penalty function is sum of squares – i.e., Minimize $M(\beta) = L(\beta|X, y) + \lambda \sum_i \beta_i^2$

Useful	▪ Allows for grouped selection and controls for multicollinearity, which provides greater model stability
	  ▪ Coefficients will be similar for two highly correlated variables
	  ▪ Coefficients will be the same for identical variables
	▪ Grouped selection can help feature creation

Not Useful	▪ Dimension reduction as it never forces a parameter to zero

Lasso (Least absolute shrinkage and selecting operator)

- Penalty function is sum of absolute values – i.e., Minimize $M(\beta) = L(\beta|X, y) + \lambda \sum_i |\beta_i|$

Useful	Understanding variable importance (as λ decreases from infinity to zero, variables start to enter model one at a time, based on importance)
Not Useful	Cannot do grouped selection (picks only one of a group of highly correlated variables)

Elastic net

- Minimize $M(\beta) = L(\beta|X, y) + \lambda_1 \sum_i |\beta_i| + \lambda_2 \sum_i \beta_i^2$
- Combines the grouping effect of Ridge with the variable selection benefit of Lasso

Decision Trees

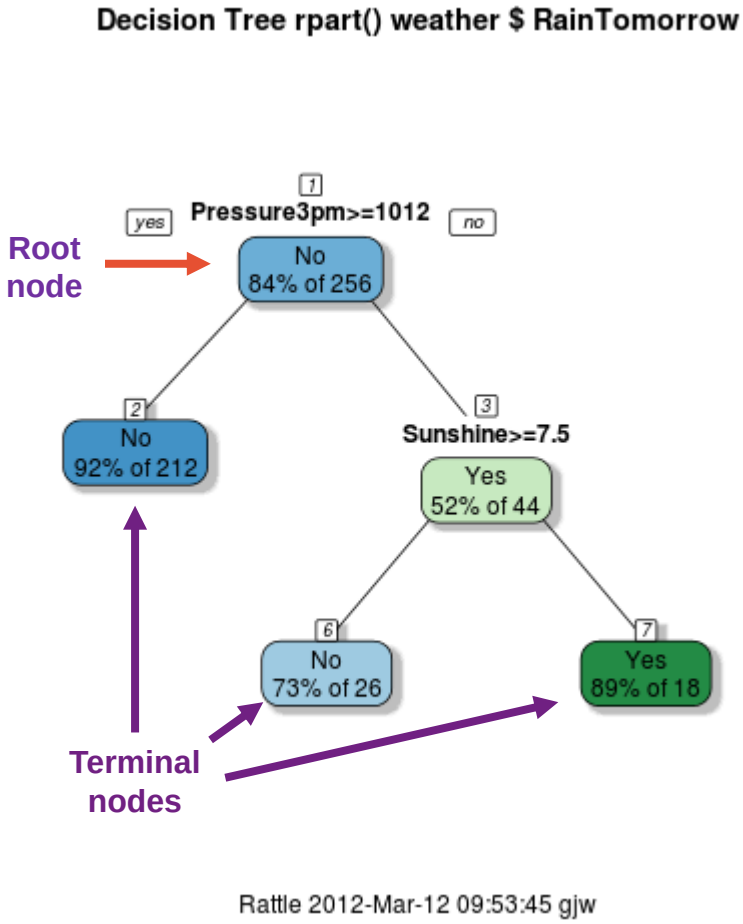
Model form

- Ask a series of yes/no questions about X to continuously split data until reach a terminal node condition
- The same prediction is made for each data point within a terminal node
- Can be regression or classification

Model fit

- Tree is grown from the root node, with each iteration splitting the data into two further sub-segments
- Optimal split at each node minimizes deviance within each node and maximizes deviance between nodes
- The algorithm stops when threshold conditions are met (e.g., exposure within the node becomes too small or maximum depth of tree has been reached)

Useful	Understanding variable importance or identifying segmentation
Not Useful	Making accurate predictions, unstable



Support Vector Machines

Model form

- Hyperplanes in a higher dimensional space **perfectly** separate data points into their response class
- New data points are predicted according to which side of the hyperplane they fall

Model fit

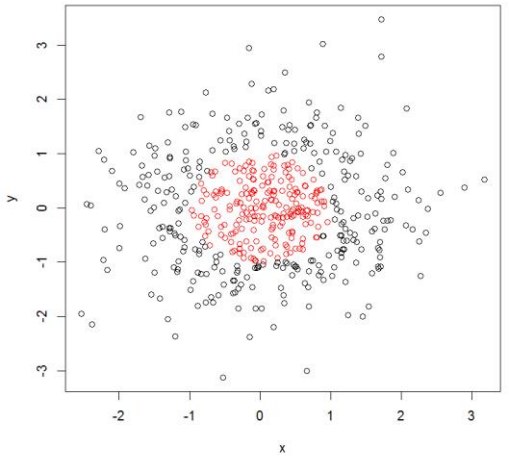
- Optimal hyperplanes maximize distance to nearest training points of each class
- "Kernel" used to *implicitly* transform data into high dimensional space where data can be linearly separated

---

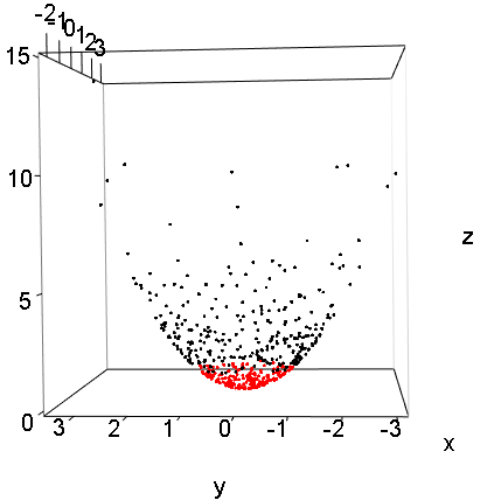
- Rarely possible to linearly separate classes perfectly in reality…
  - Add a small error tolerance for to handle noise

Useful	Accuracy; picks up signal in higher dimensions
Not Useful	Interpretation, may be unstable

Original feature space



After transformation



Neural Networks

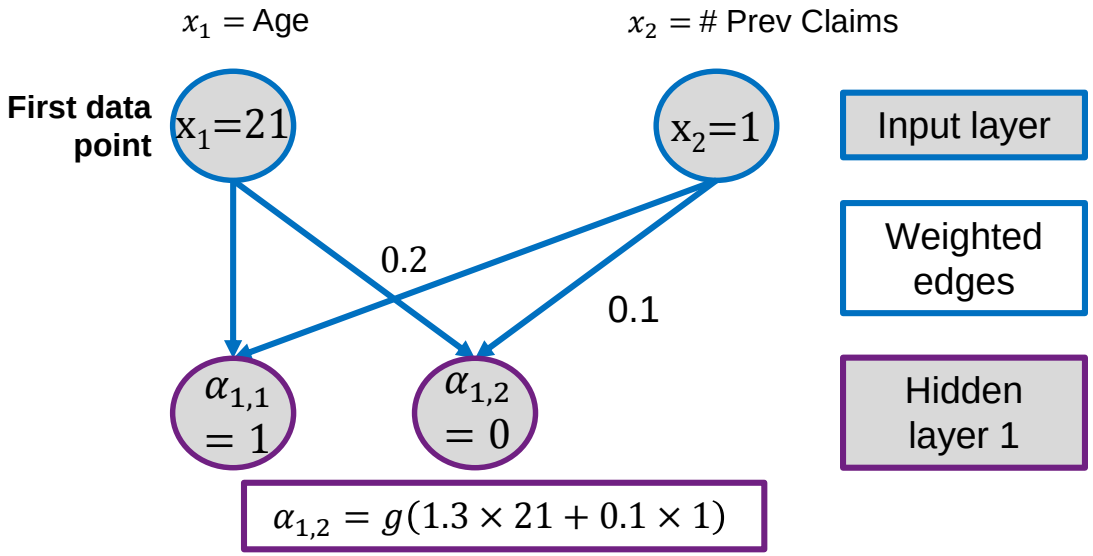
Model form

- Predicted values calculated in stages by passing through a network of layers
- Value of a node given by a **weighted sum** of values in the layer before…
- …transformed by a non-linear **activation function**, $g(\cdot)$
- Choose the number/size of hidden layers

Model fit

- Weights estimated to minimize a loss function (e.g., negative log-likelihood)

Useful	Accuracy, particularly for image data (less so for structured data)
Not Useful	Interpretability



Neural Networks

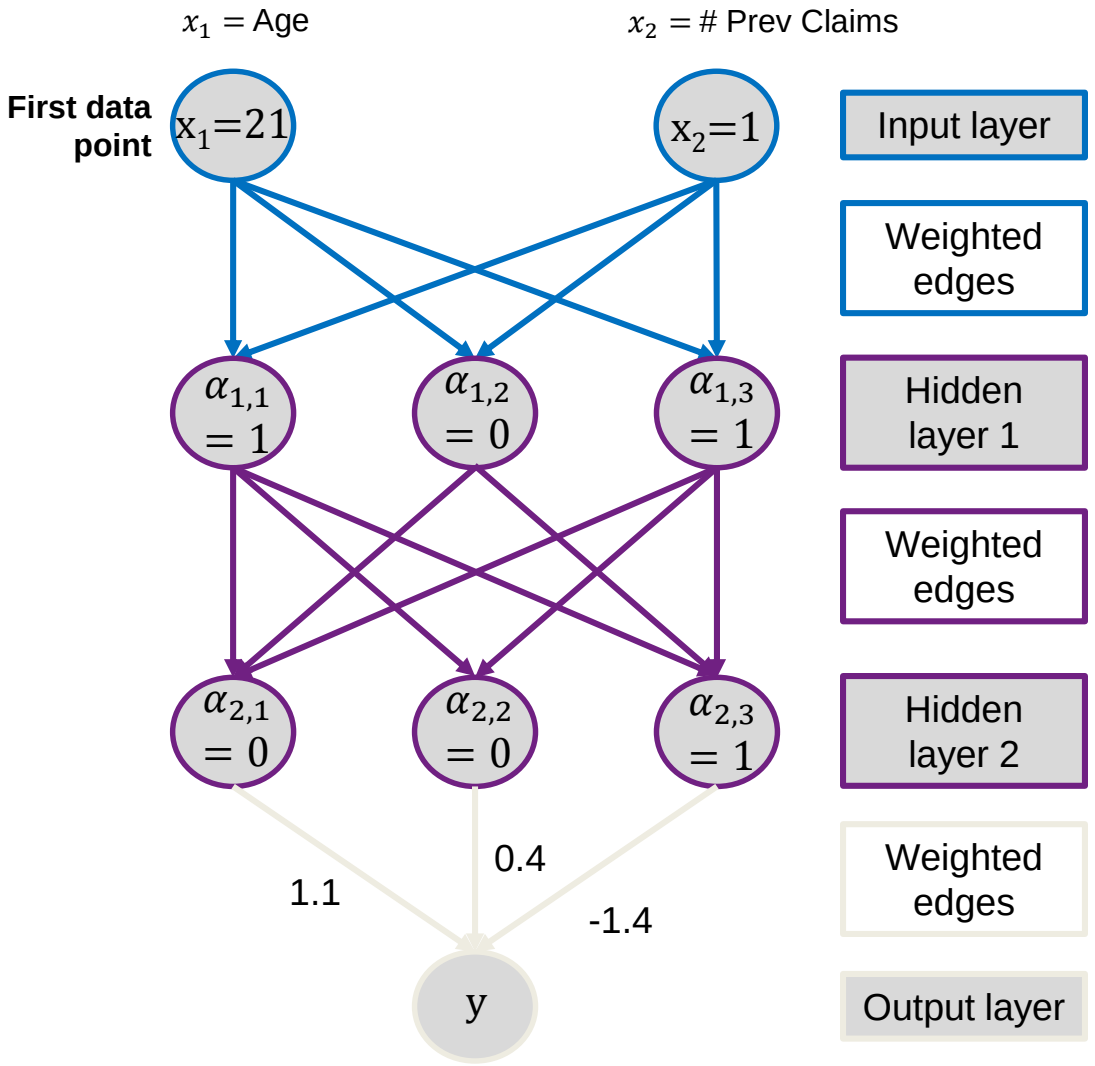
Model form

- Predicted values calculated in stages by passing through a network of layers
- Value of a node given by a **weighted sum** of values in the layer before…
- …transformed by a non-linear **activation function**, $g(\cdot)$
- Choose the number/size of hidden layers

Model fit

- Weights estimated to minimize a loss function (e.g., negative log-likelihood)

Useful	Accuracy, particularly for image data (less so for structured data)
Not Useful	Interpretability



Check answer against test data, calculate gradient of loss function and move back through the network to update weights

$$y = g(1.1 \times 0 + 0.4 \times 0 - 1.4 \times 1)$$

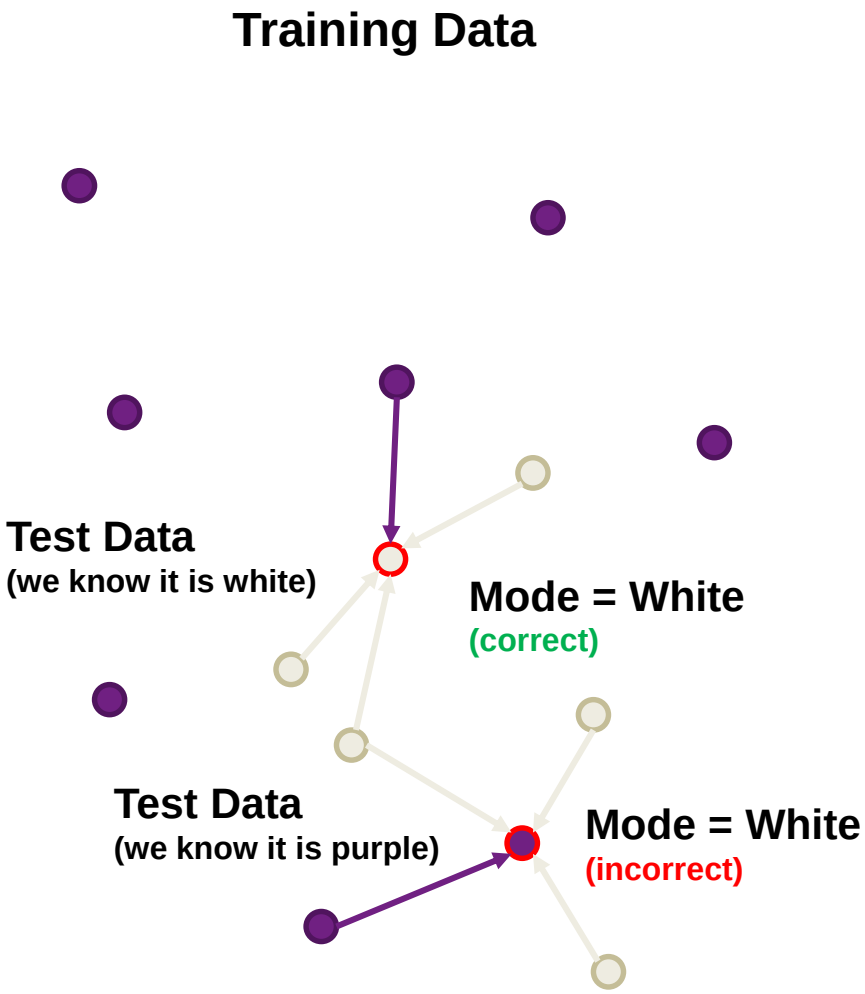
K-Nearest Neighbors

Model form & fit

- Training data and test data plotted in n-dimensional space
- Predictions made for test data by identifying the k nearest neighbors from the training data and taking their average response
  - Modal response for classification
  - Mean response for regressions

- K chosen for maximum accuracy

Useful	When data and application allows for empirical solution, stable method
Not Useful	For gaining insights about the data or the underlying process



Continue with all test data to calculate average prediction error

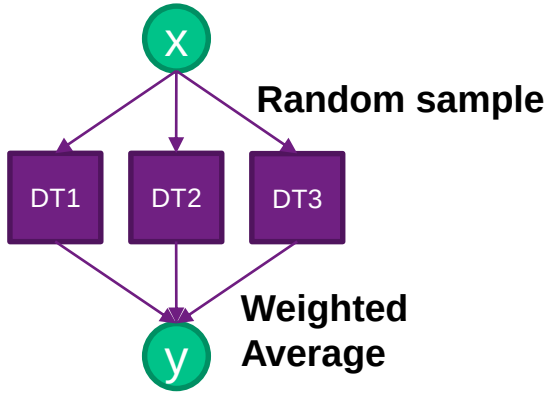
Ensembles

Ensembles combine (or blend) information from various models in order to improve the prediction

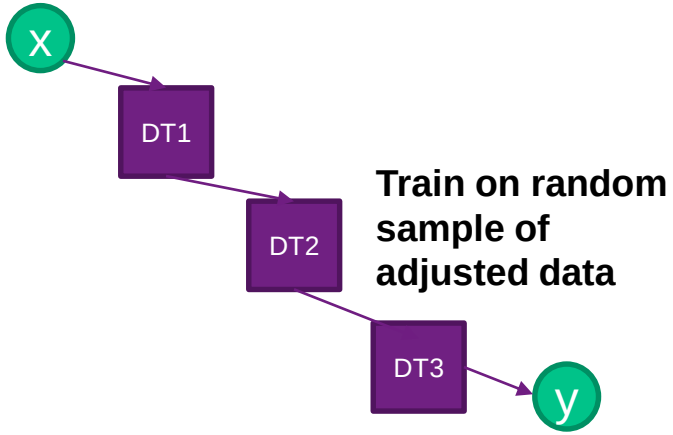
- Bagging (**b**ootstrap **ag**gregation) aims to reduce the variance of the prediction
  - Train models (learners) independently and take a weighted average of results
  - Same model type is applied each time, but to different randomly sampled versions of data (observations are sampled with replacement to achieve equal size datasets)
  - Example: random forests
- Boosting aims to reduce the variance and the bias of the prediction
  - Each model (learner) is trained on how the previous model performed (i.e., not an independent process)
  - In subsequent learners, more weight is given to observations that didn't validate well previously
  - Example: gradient boosting machines (GBM)

Useful	For allowing complexity without overfitting
Not Useful	For interpreting final result; may be challenging for insurers to operationalize

Bagging



Boosting



Random Forests (an example of bagging)

Model form

- Large number of Decision Trees, each giving an independent prediction
- Overall prediction is a weighted average prediction across all trees
  - Regression models use the **mean**
  - Classification models use the **mode**
- Each tree's prediction weighted by its predictiveness

Model fit

- Each tree is trained independently of the others, using:
  - A random subsample of the available factors (so that no single factor can dominate every tree)
  - A **bootstrapped** random sample of the training data (to prevent over-fitting to the training data)

Useful	Variable selection (importance) and model stability; highly scalable
Not Useful	Predictive accuracy (compared to boosting ensembles)

Gradient Boosted Models (an example of boosting)

Model fit

- The overall prediction is given by:

$$f(x) = \lambda \sum_{n=1}^{N} f_n(x)$$

- Base models are usually Decision Trees, but could use other model forms (e.g., GLMs)

Model form

- Models are successively trained on the **residuals** of the previous model
- At each iteration, the model is updated by adding a fraction (λ, or the shrinkage) of the new model
- Each subsequent model $f_m(x)$ is fit to the residuals, e.g.,: $y - \lambda \sum_{n=1}^{m-1} f_n(x)$
- As with Random Forests, each iteration performed on a random sample of factors and data points to reduce over-fitting to the training data

Useful	For accurate prediction by incorporating complex non-linear relationships
Not Useful	Not useful when interpretability is important; may also be challenging to implement

K-Means Clustering

Model form

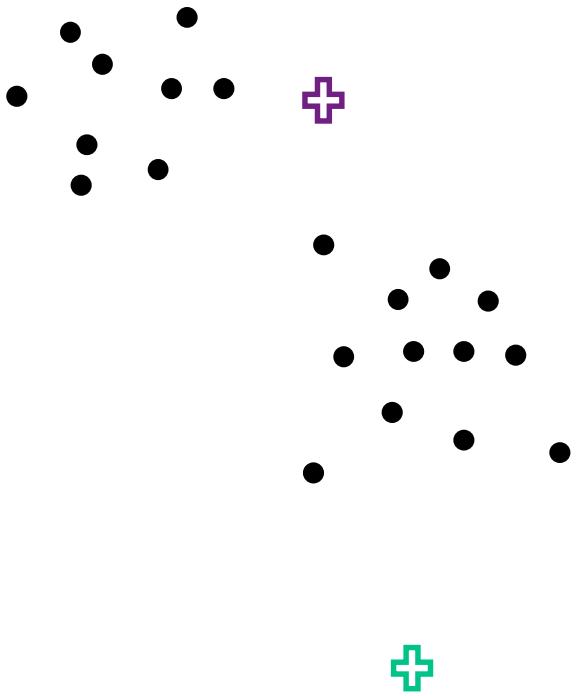
- An example of **unsupervised learning** (no response variable)
- Data points are assigned a class based on the nearest of k “centroids”

Model fit

- Centroids are selected to minimize the distance measure from data points to their nearest centroid

- Note the distinction between **clustering** (unsupervised) and **classification** (supervised)…

Useful	For clustering similar things
Not Useful	For predictions or when no obvious measure of similarity



Kaggle winning methods (January 2015 to February 2016)

- **Gradient Boosted Machines** was most successful technique across the board
- **Feature Creation/Selection** was noted as biggest contributor to success
  - The nature of Kaggle and the sharing of benchmarks means most competitors use the same algorithms – thus the key differentiator is the improvement gained from good feature creation/selection

Count of method placing "top 3" in competition (for which data was available)

Competition subject	Support Vector Machine	Gradient Boosted Machine	Neural Network	Mixed Method Ensemble	Random Forest	Total
All	1	19	10	10	1	41
Insurance	-	3	-	4	-	7

Using & evaluating methods in practice

Recap: Gradient Boosted Models

Model fit:

- **Boosting** is where models are successively trained on the **residuals** of the previous model
- At each iteration, the model is updated by adding a fraction (λ, or the shrinkage) of the new model
- Each $f_m(x)$ is fit to the residuals, eg: $\mathrm{y} - \lambda \sum_{n=1}^{m-1} f_n(x)$
- Each iteration performed on a random sample of data points to reduce over-fitting to the training data

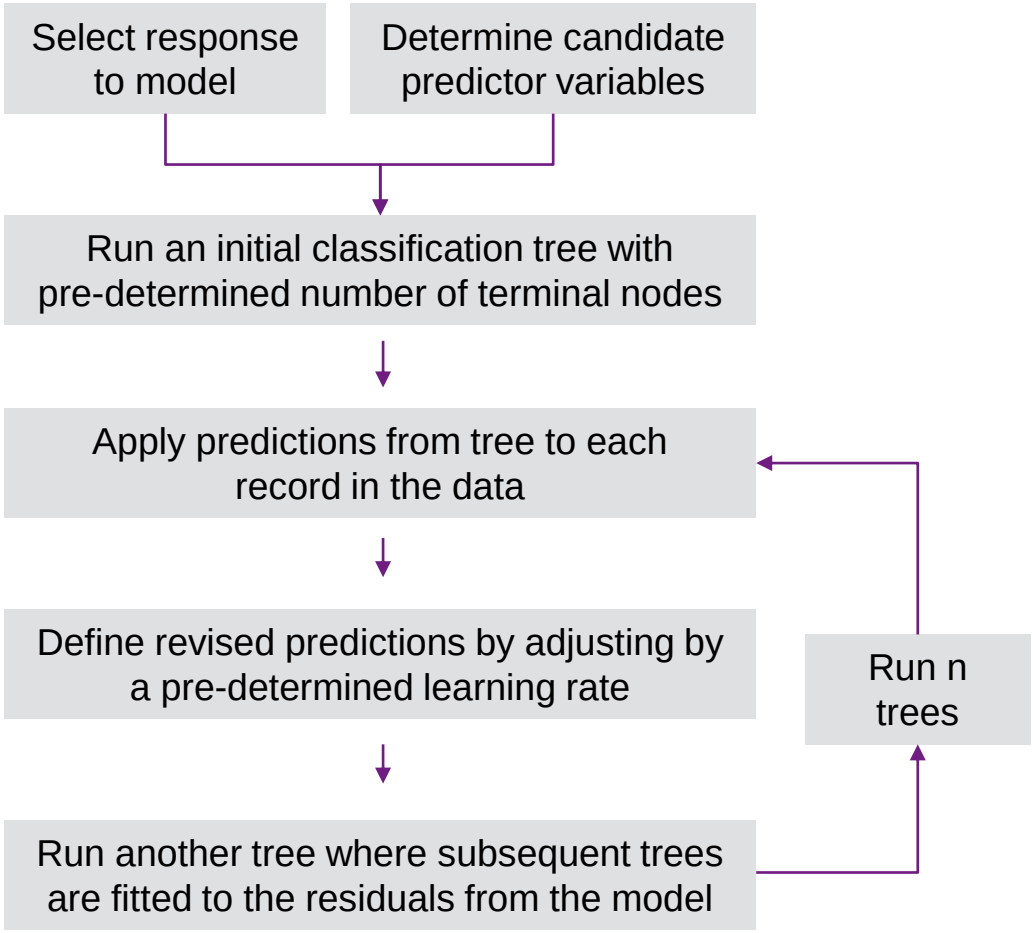
Model form:

- The overall prediction is given by

$$f(x) = \lambda \sum_{n=1}^{N} f_n(x)$$

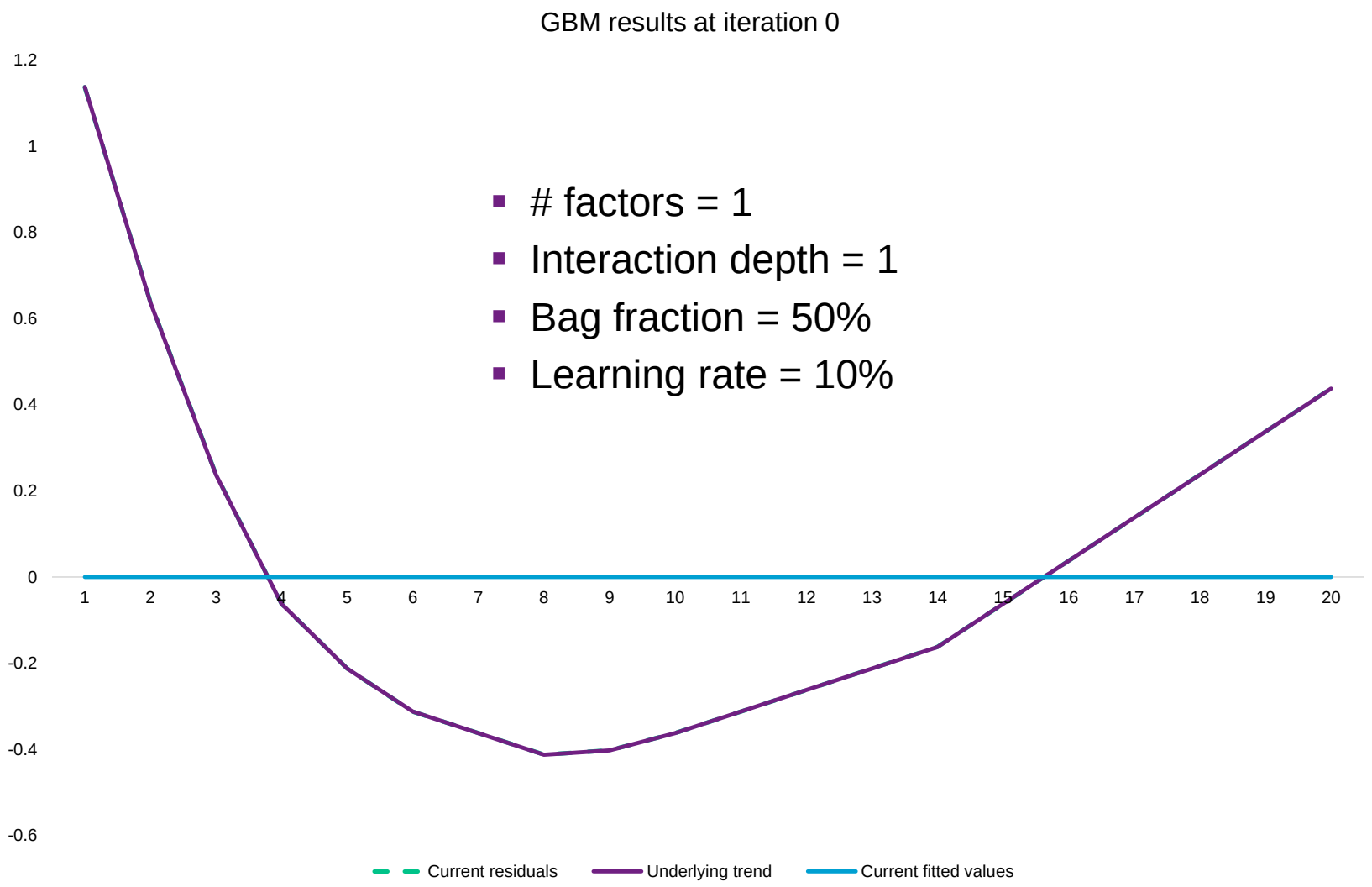
- Base models are usually Decision Trees, but could use other model forms (eg simple GLMs)

A simplistic view of the method & the four main assumptions

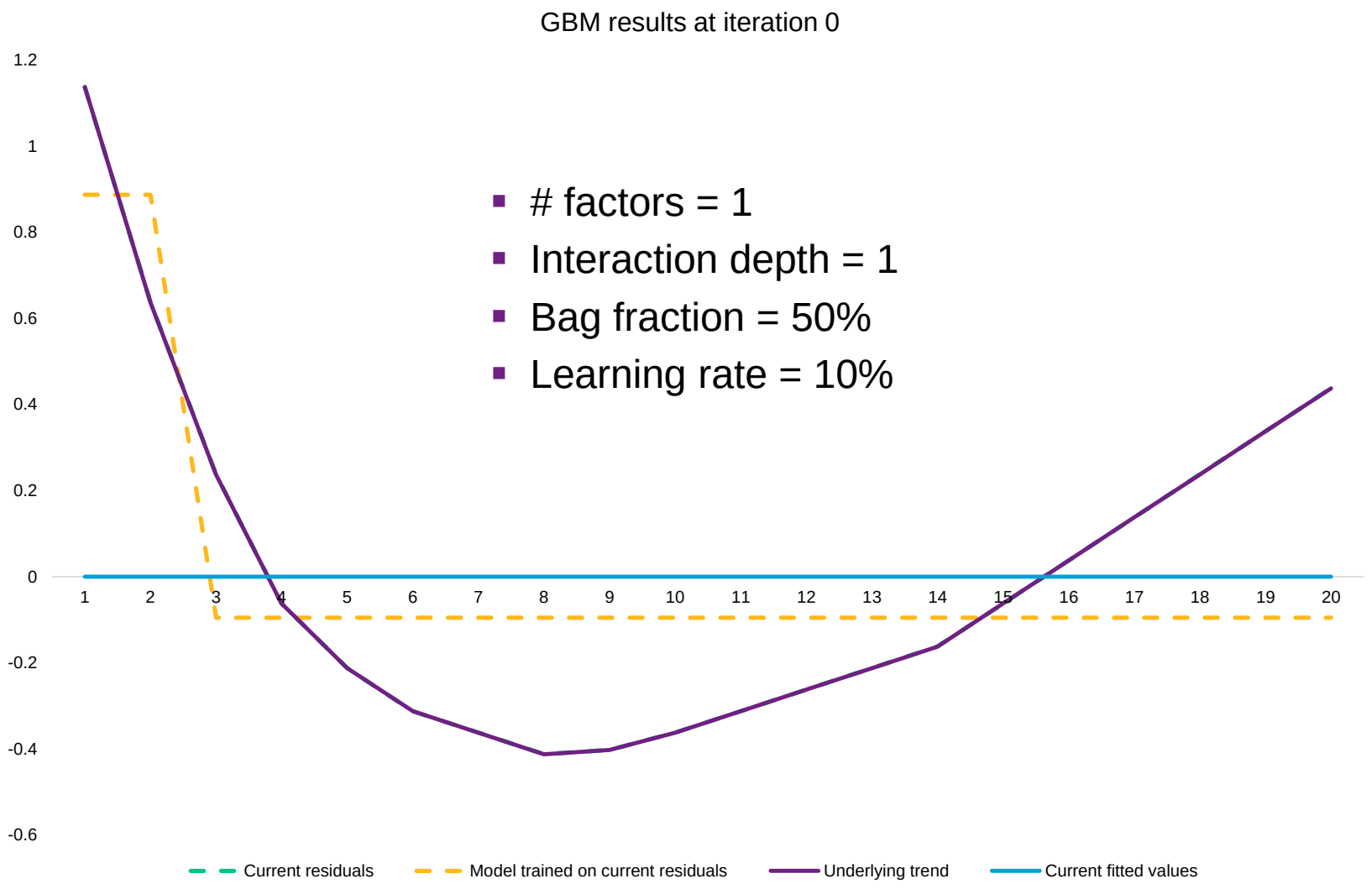


- **Learning rate / "shrinkage"**
  - Amount by which the old model predictions are varied for the next model iteration
  - New model = Old + (Prediction x Learning rate)
- **Interaction depth**
  - Number of splits allowed on each tree (or the number of terminal nodes – 1)
- **Number of trees** (iterations) allowed
- **Bag fraction**
  - Trees are fitted to a subset of the data (the bag fraction) on a randomized basis
  - Additional noise-reduction can be achieved by using a random subset of the available factors at each iteration

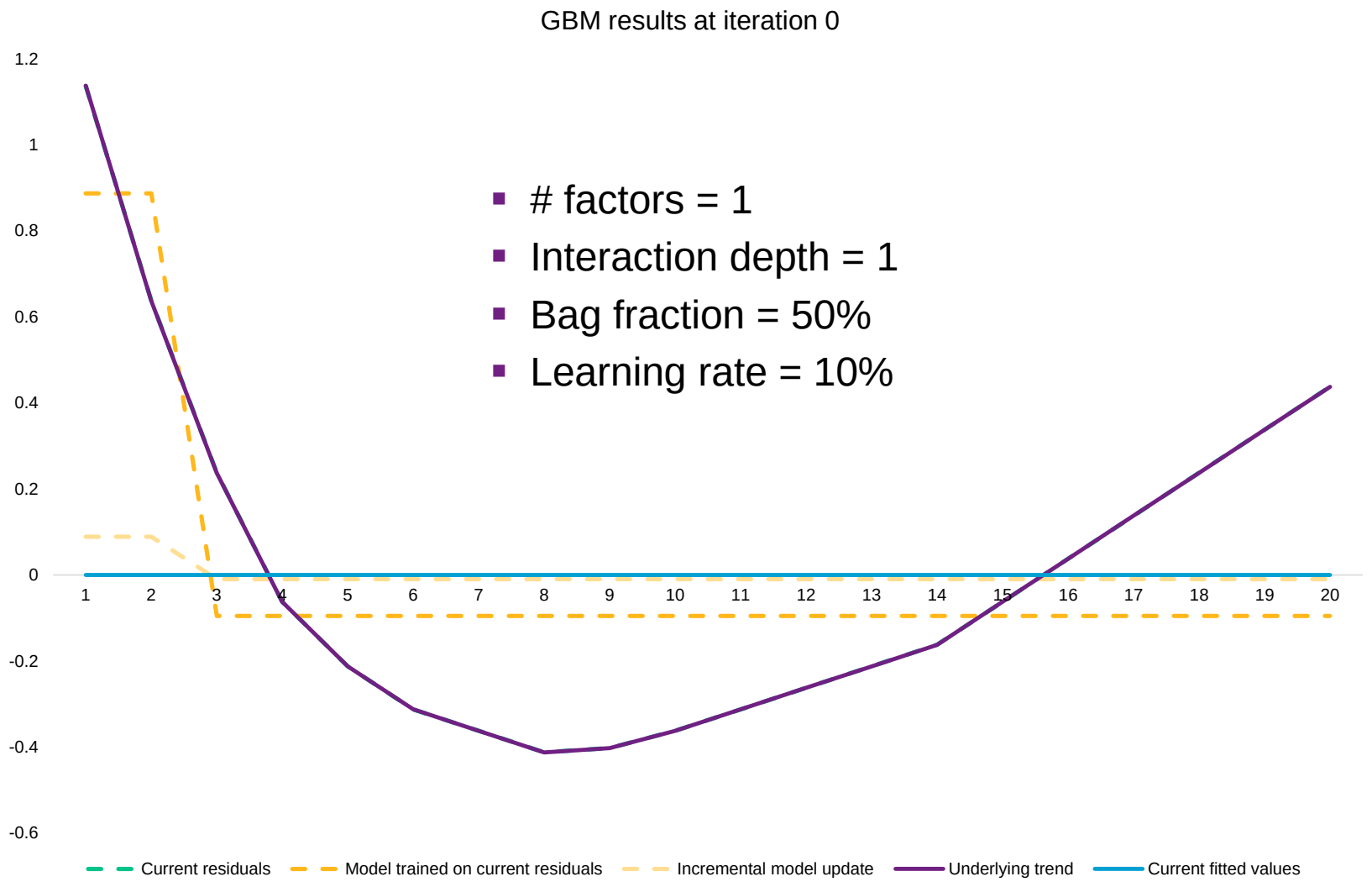
A simple example



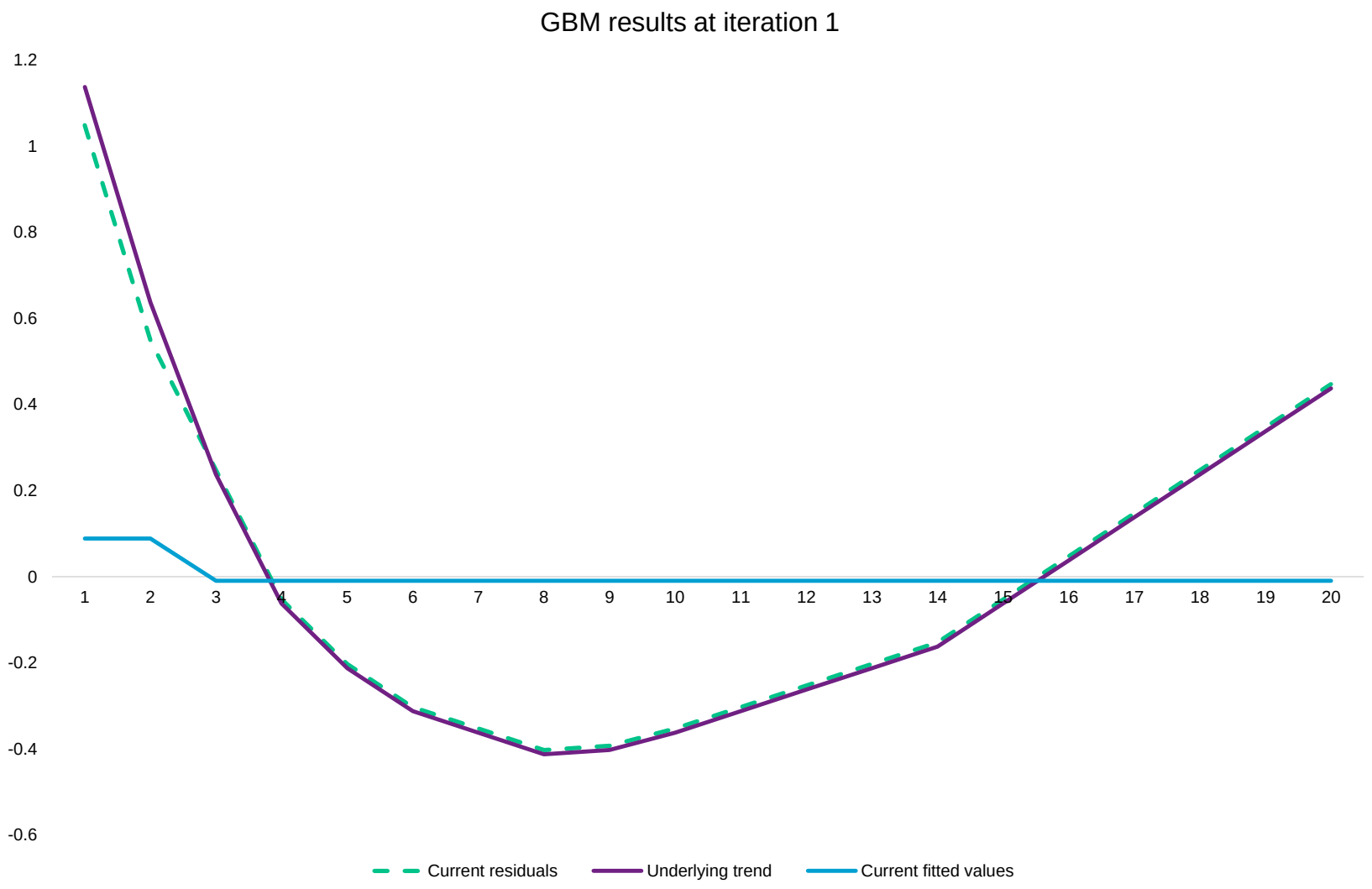
A simple example



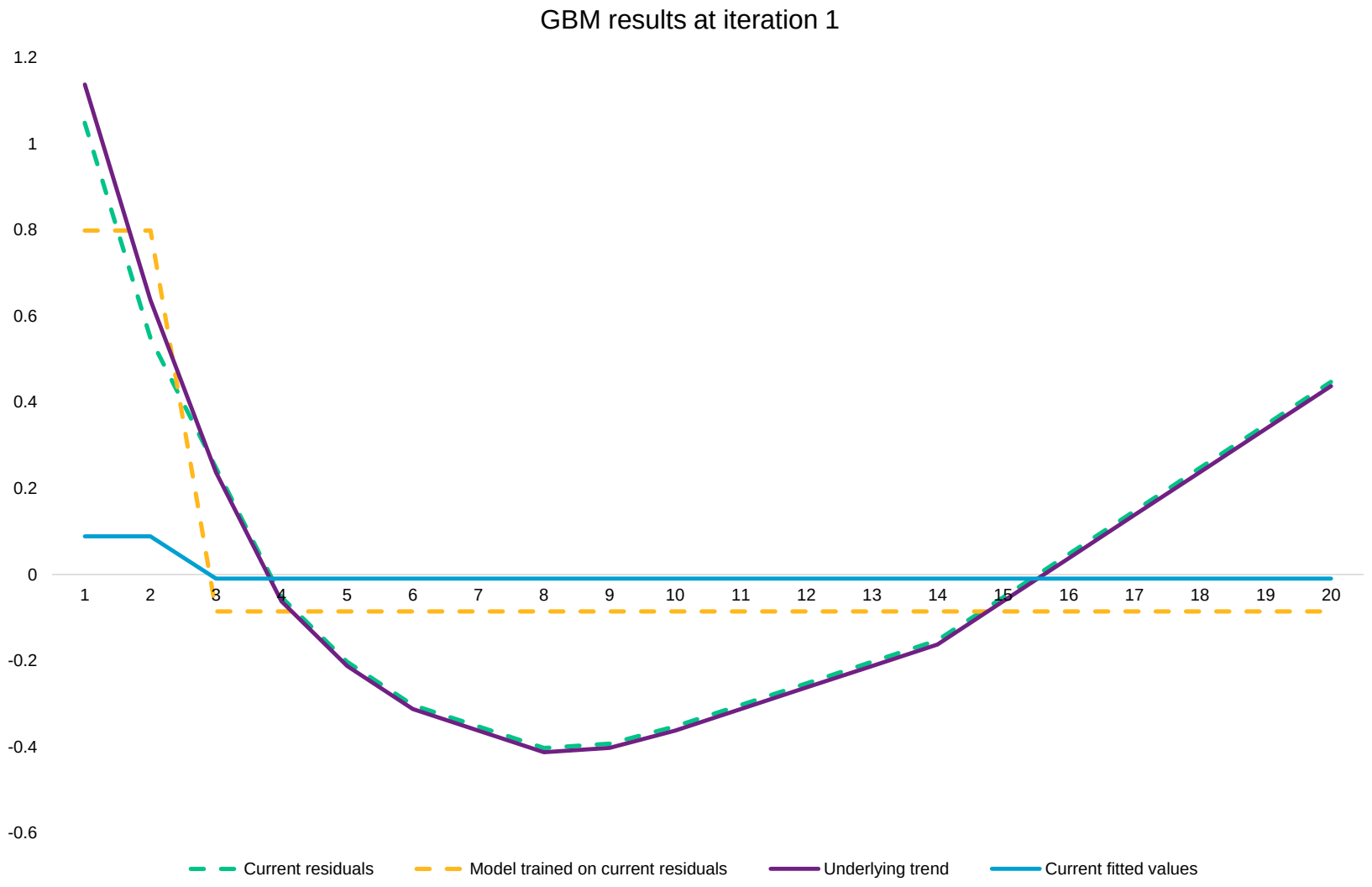
A simple example



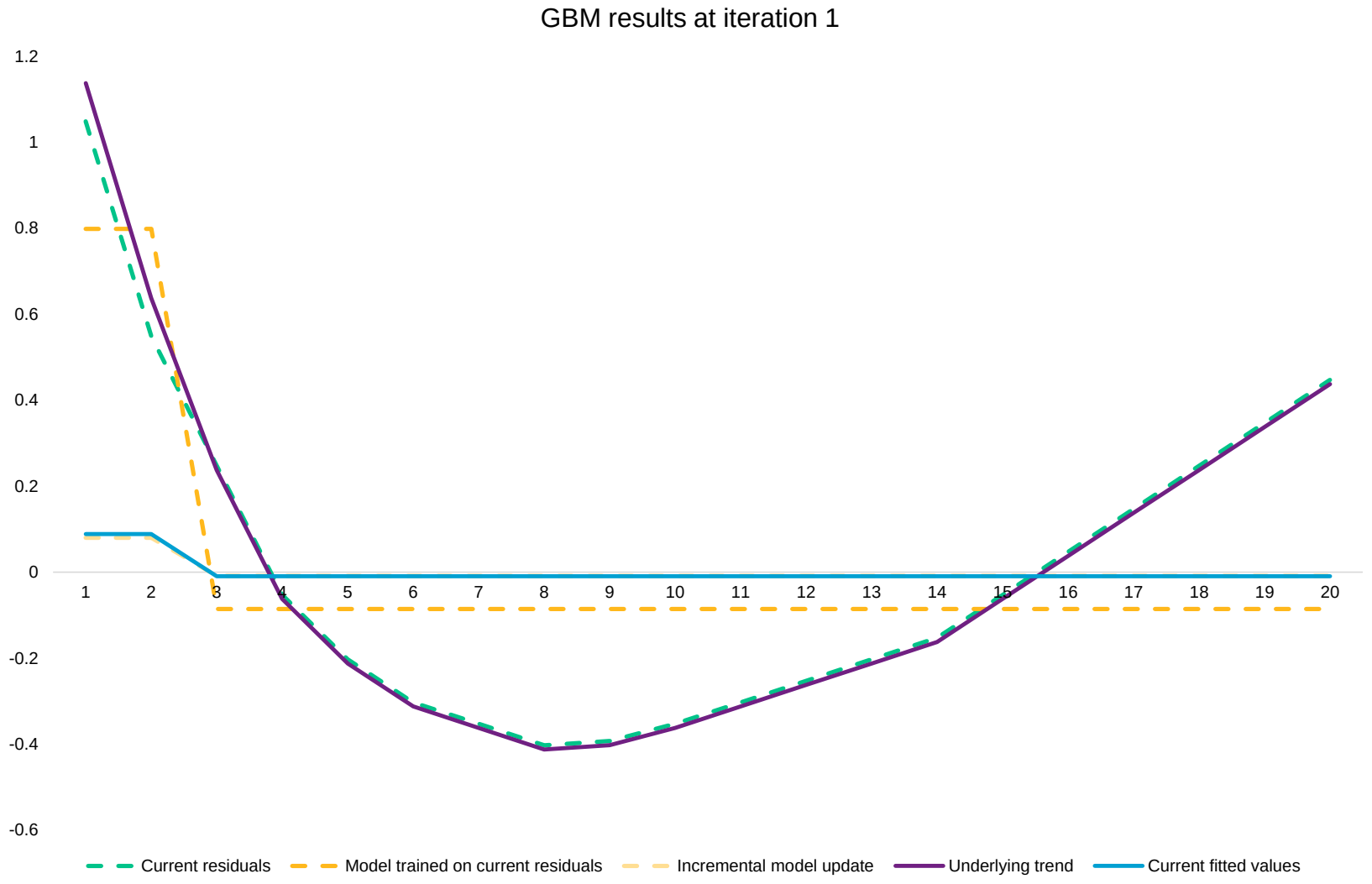
A simple example



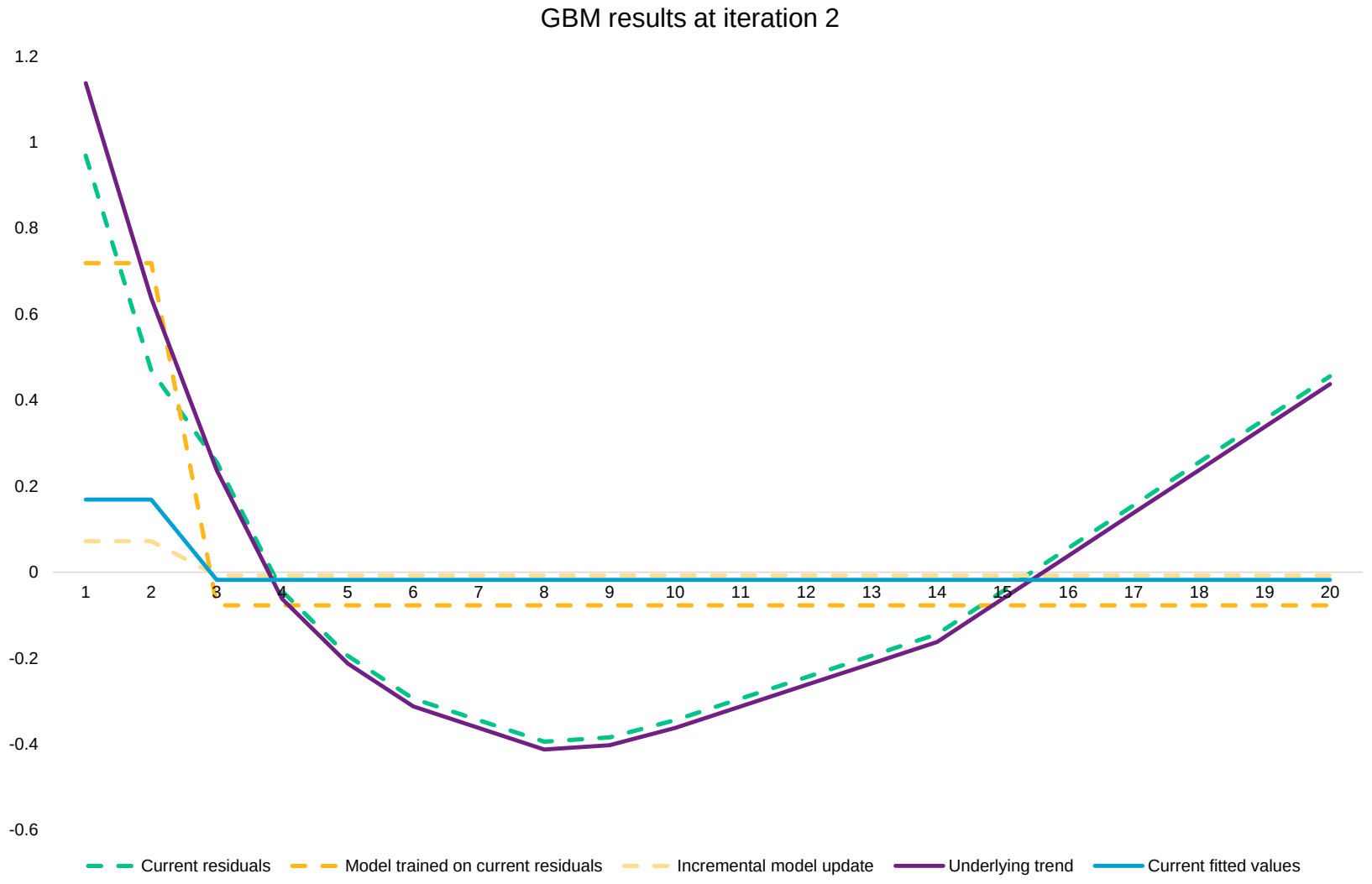
A simple example



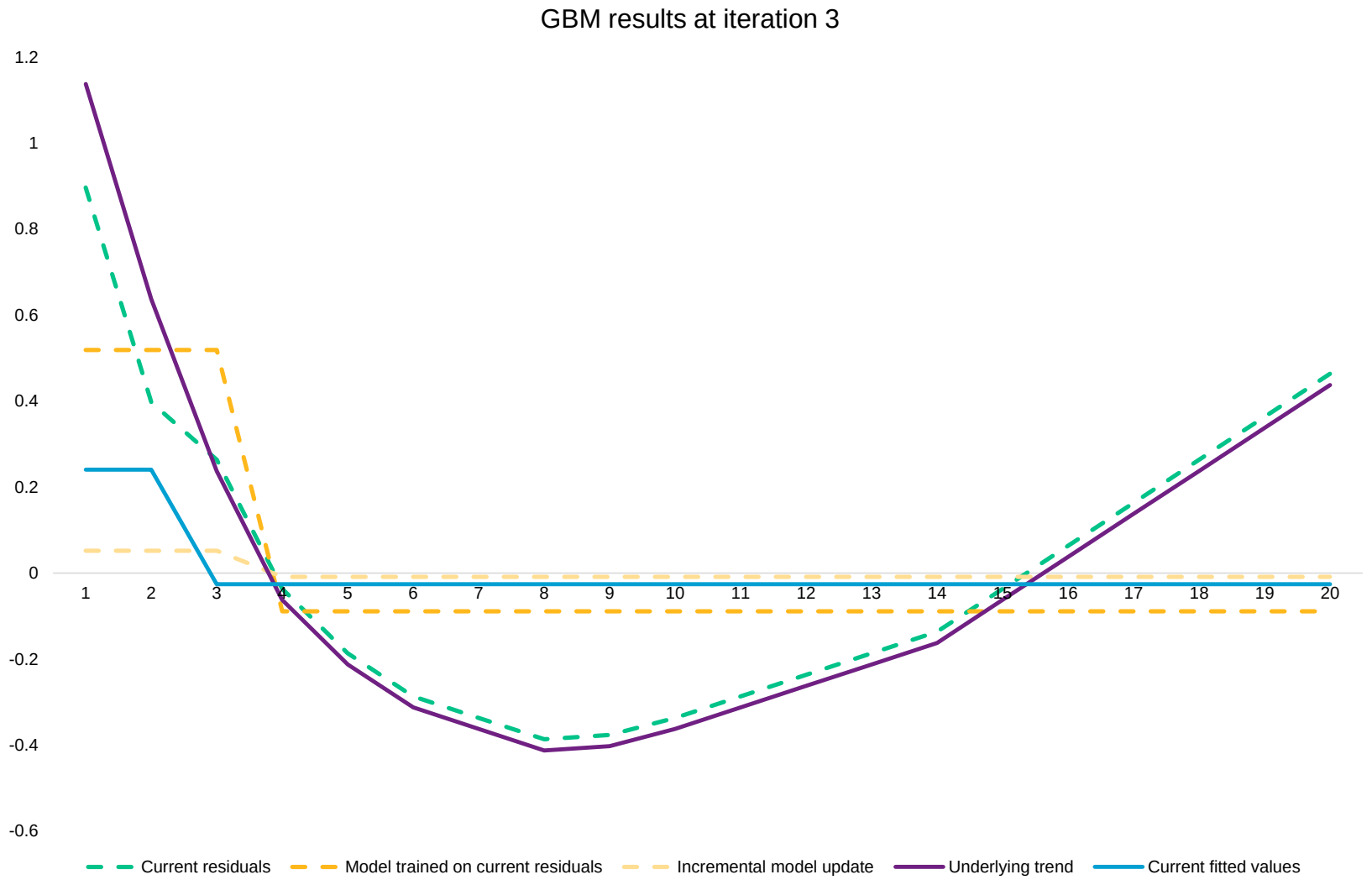
A simple example



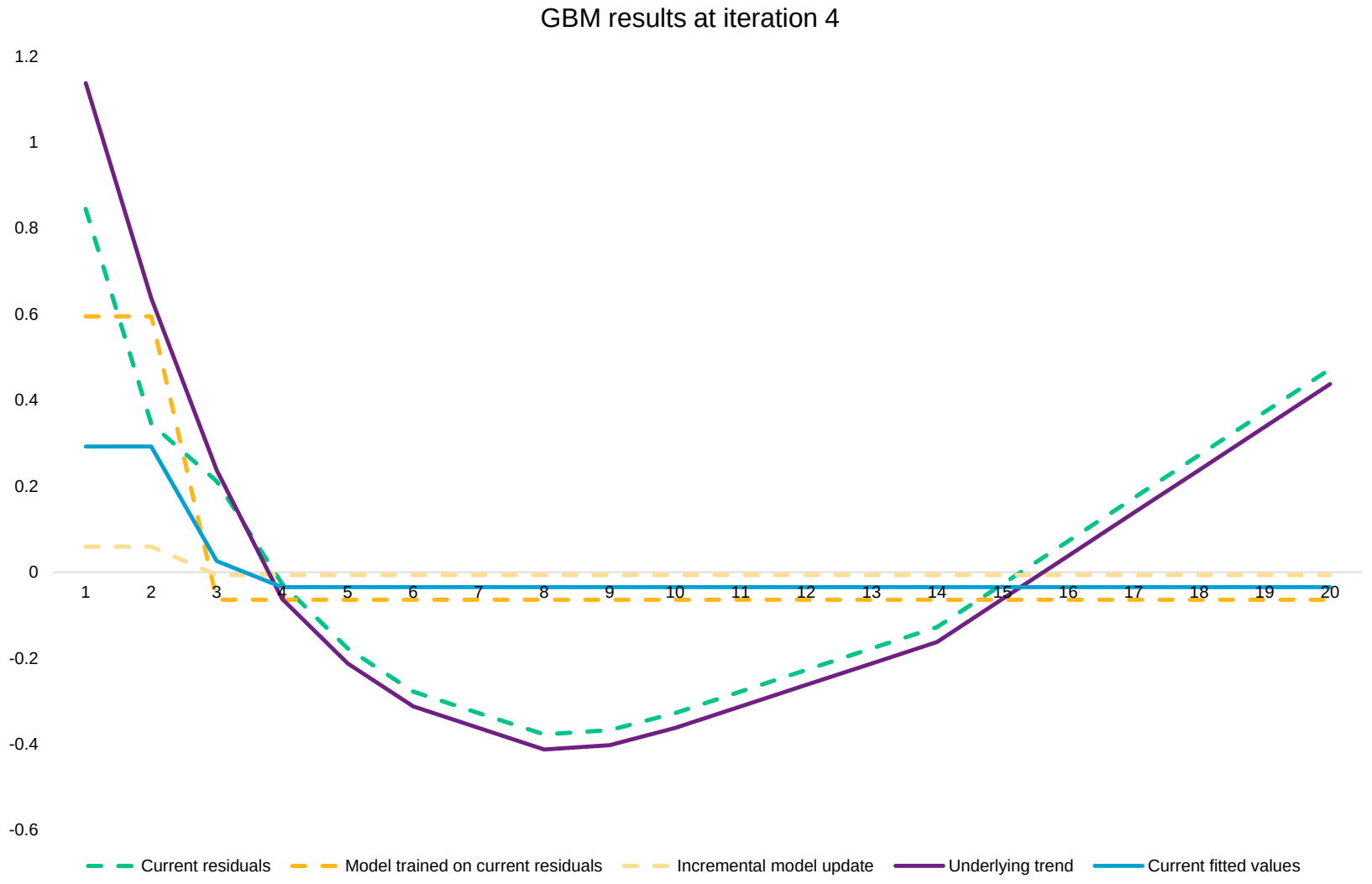
A simple example



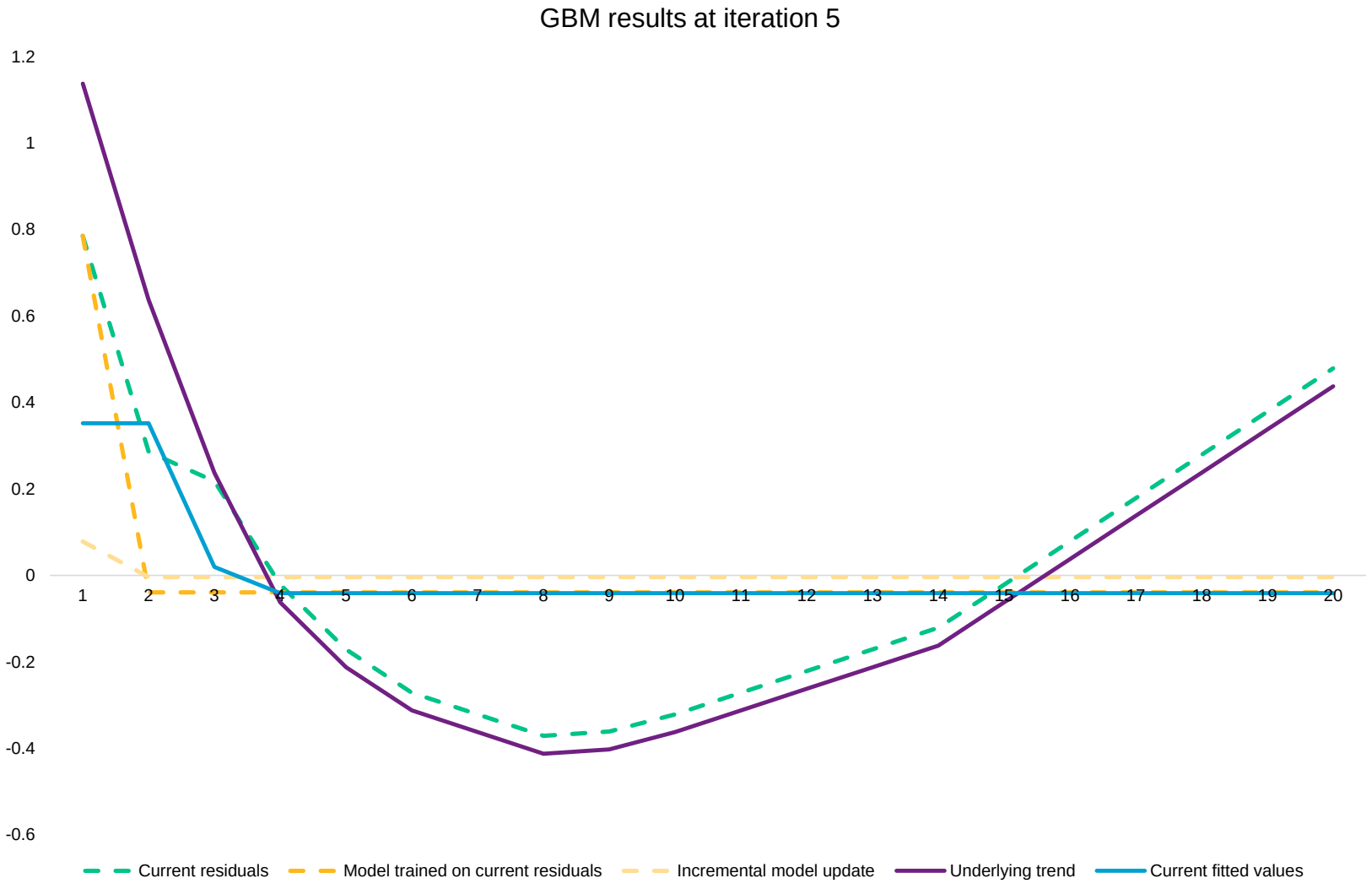
A simple example



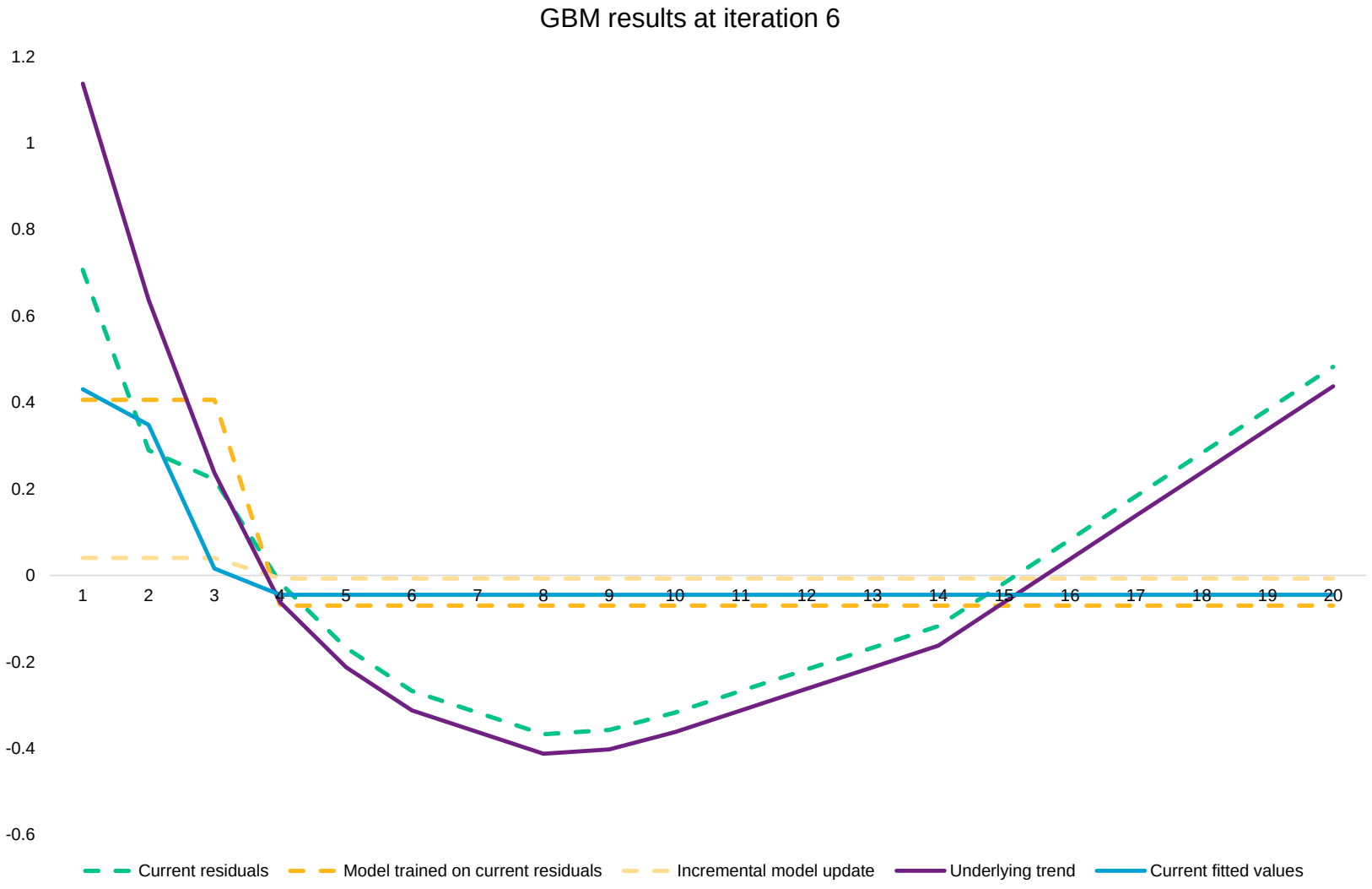
A simple example



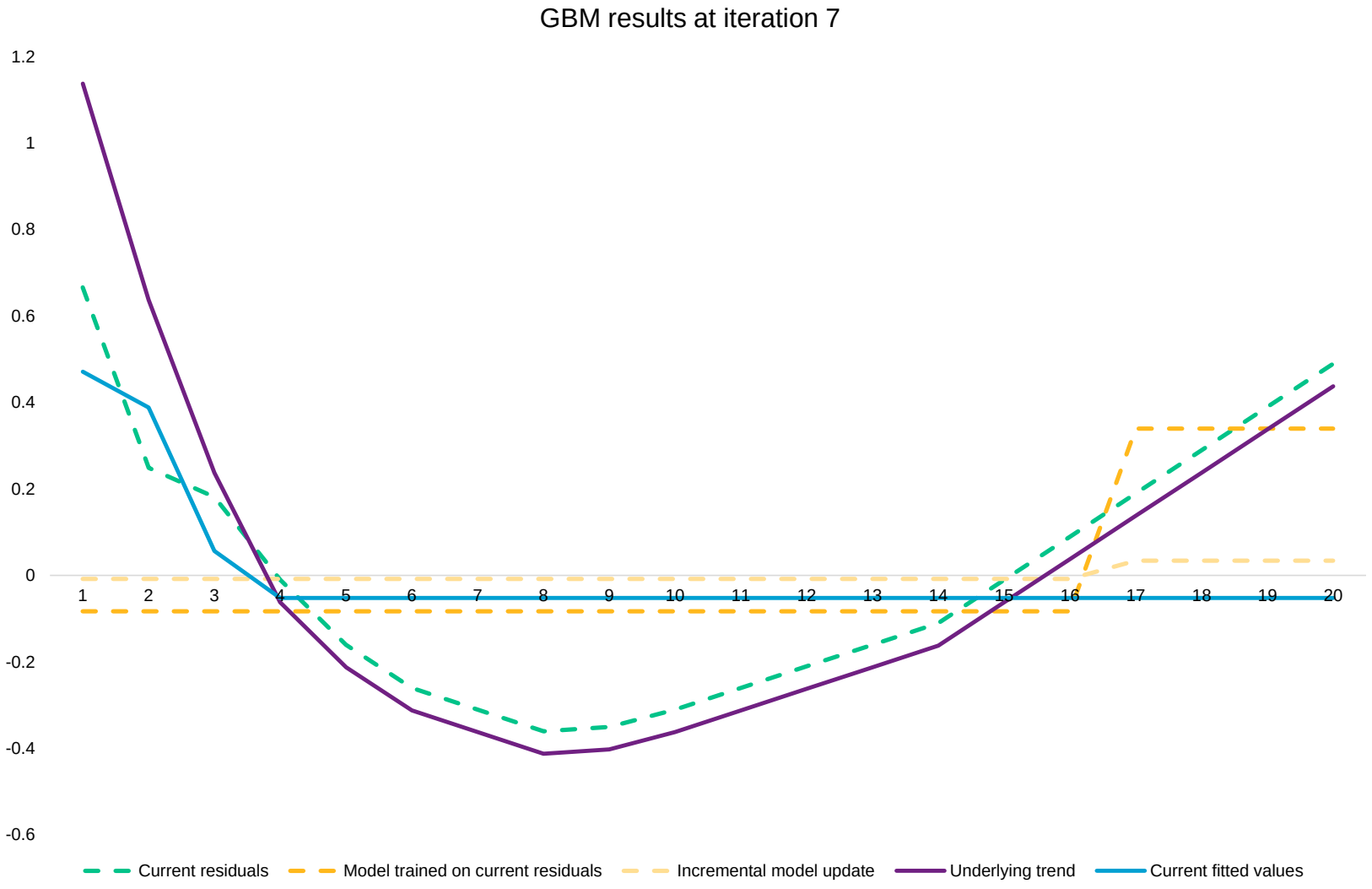
A simple example



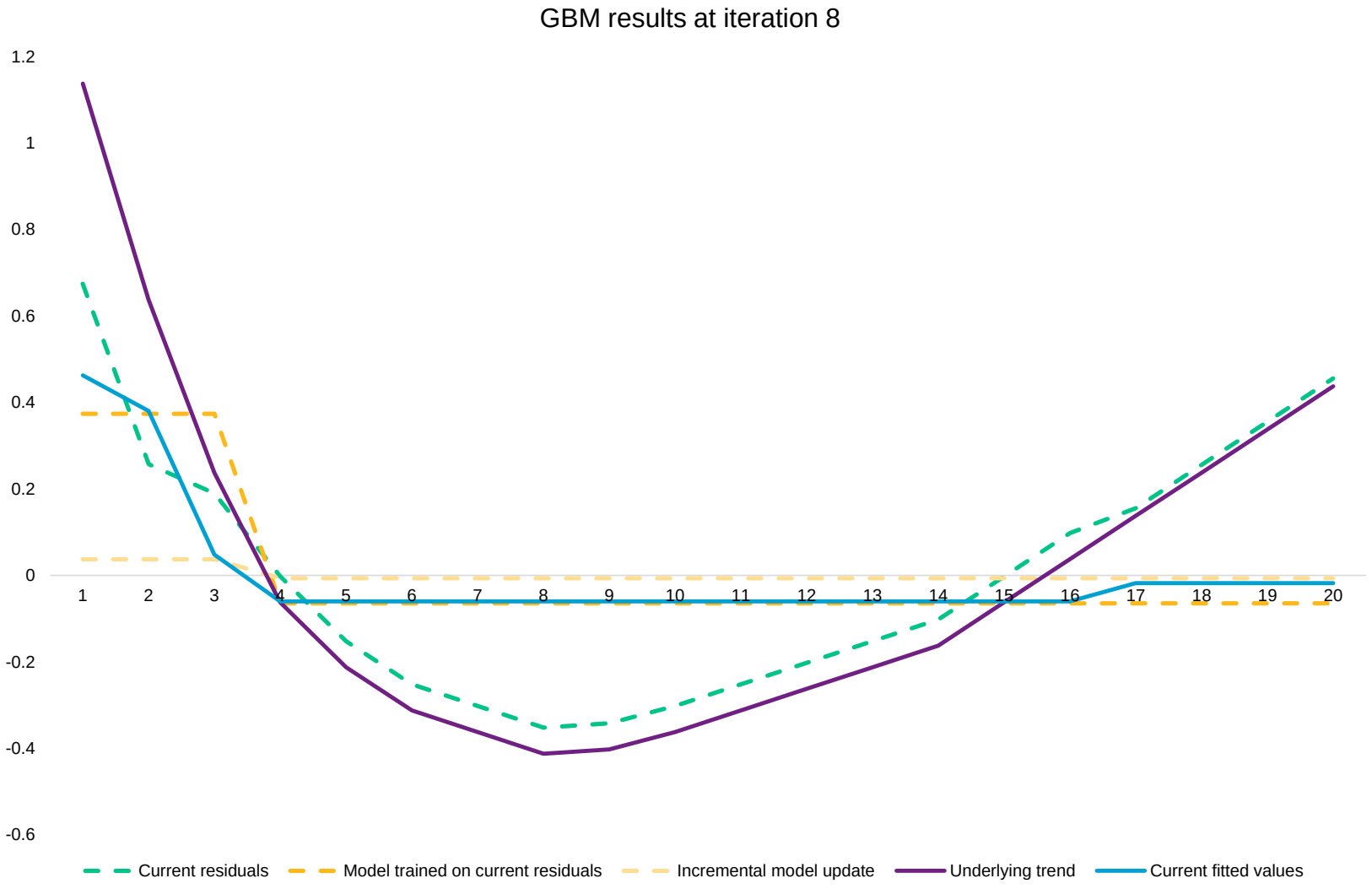
A simple example



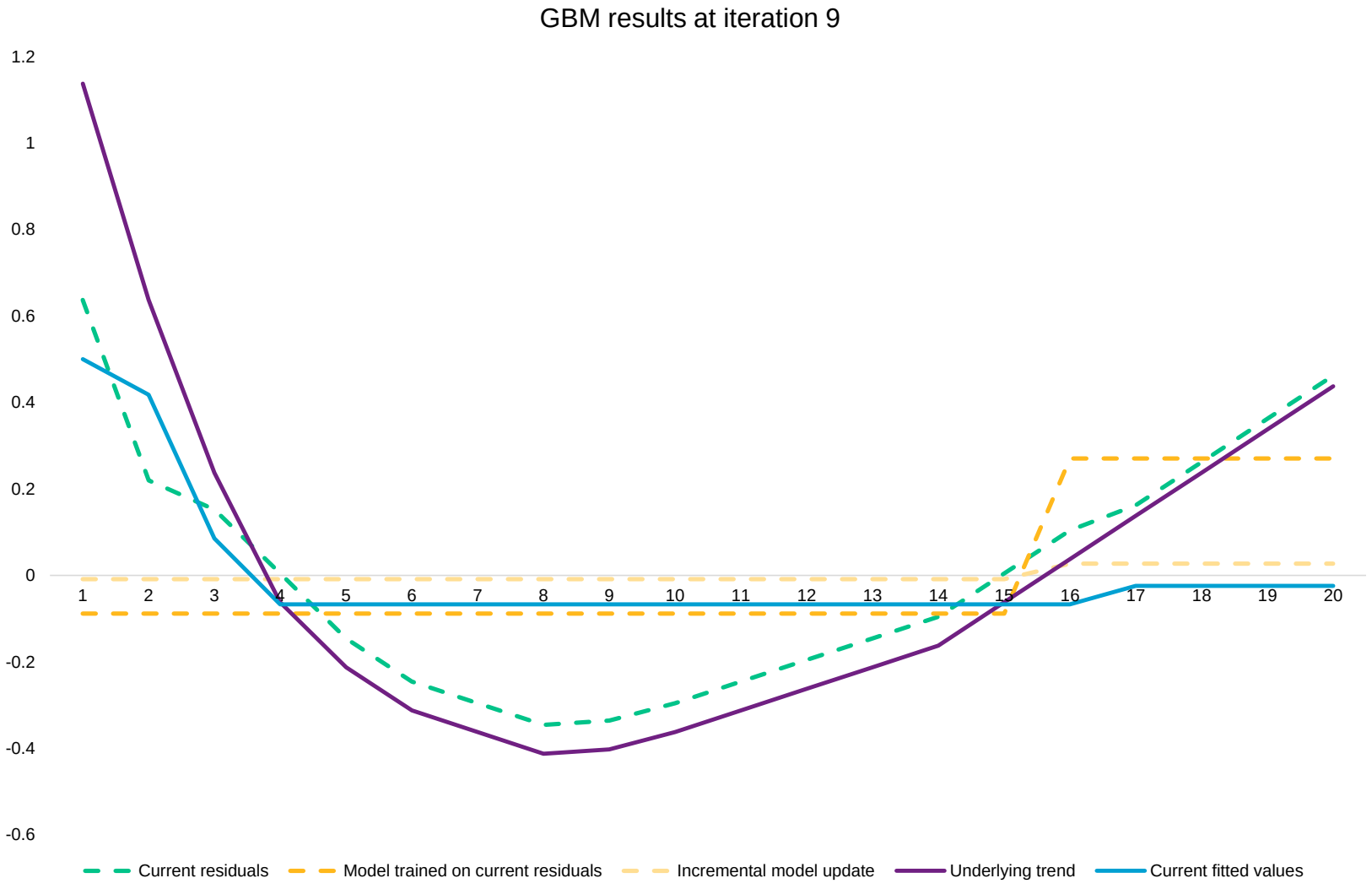
A simple example



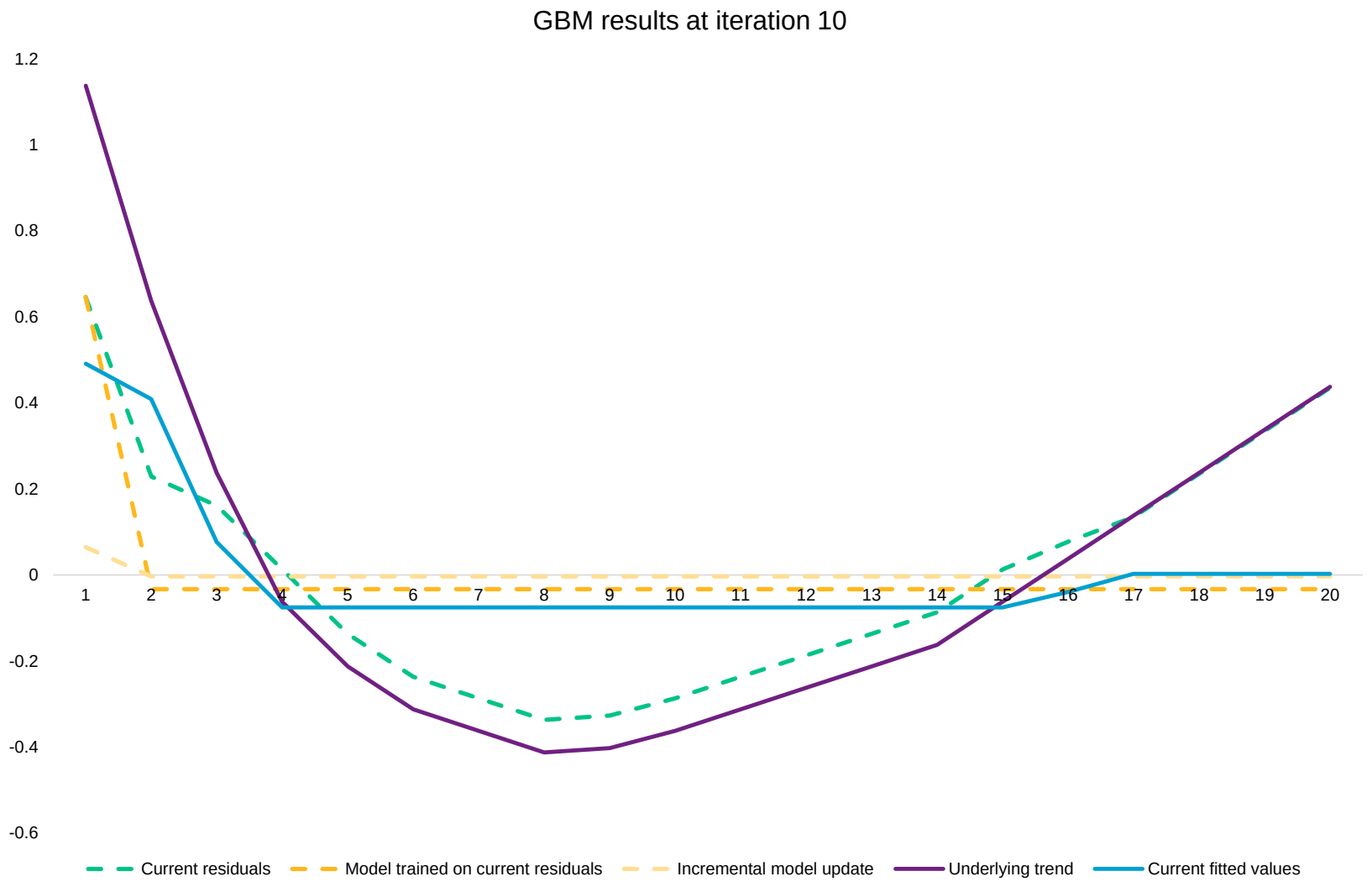
A simple example



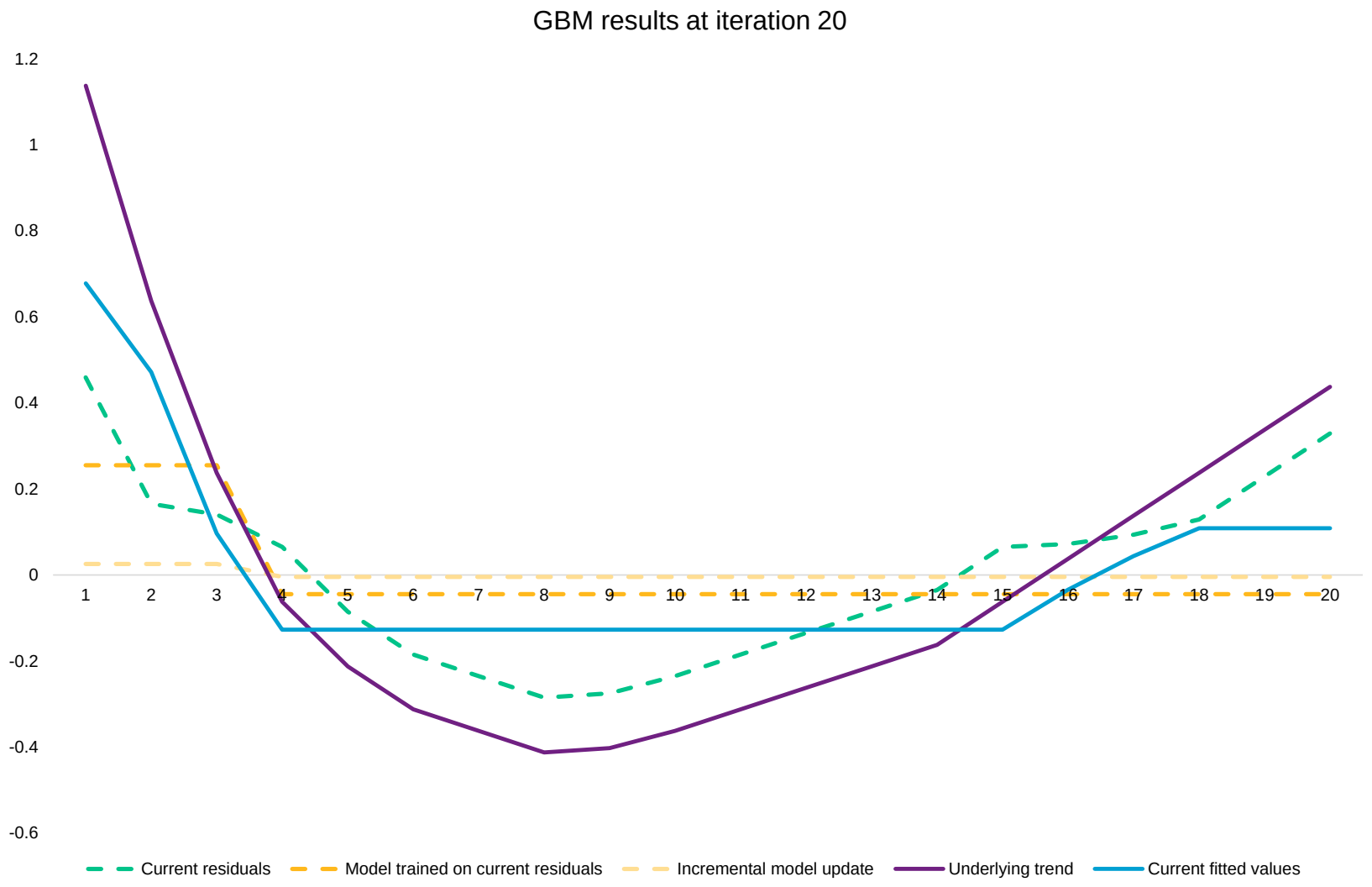
A simple example



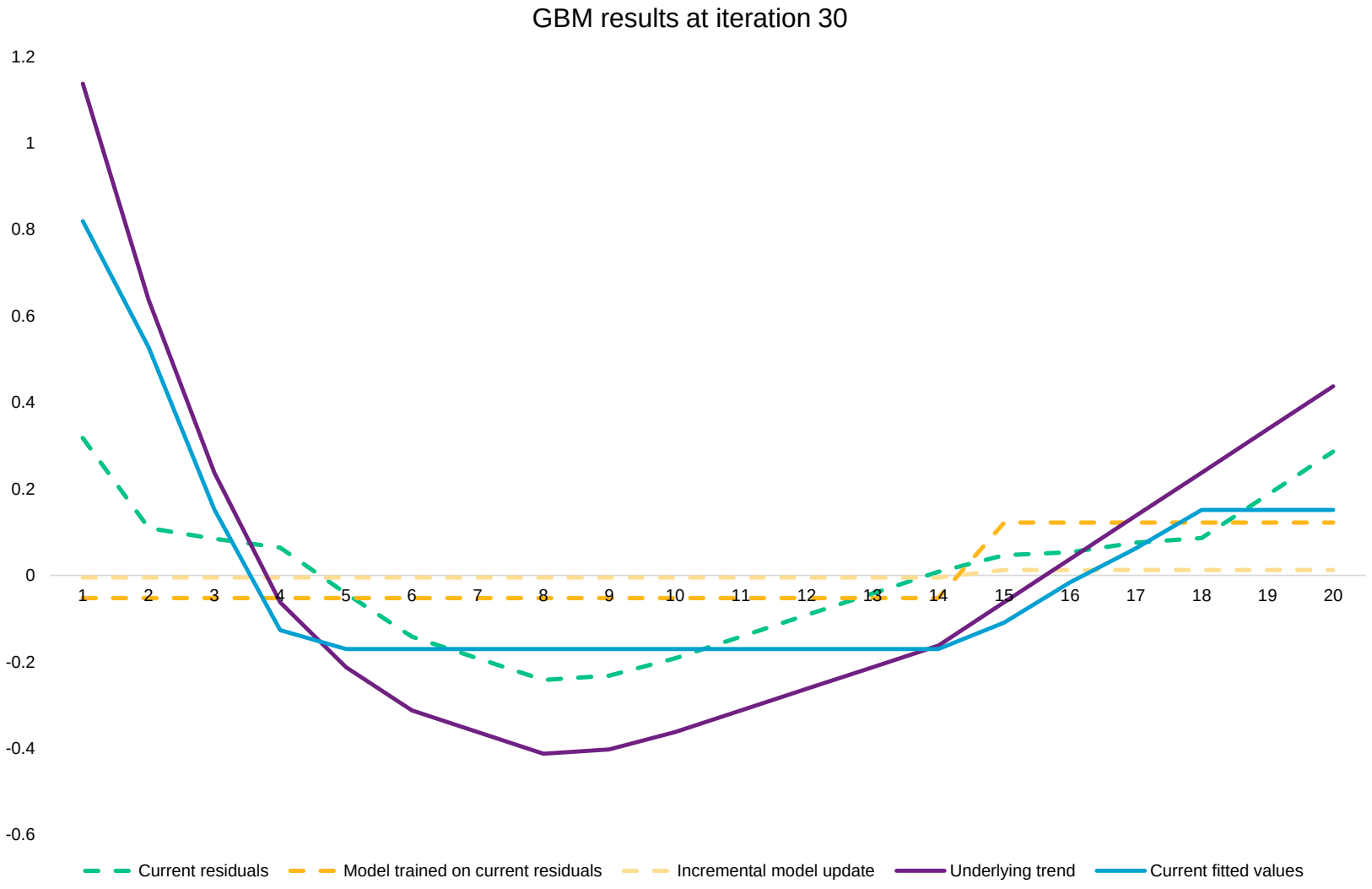
A simple example



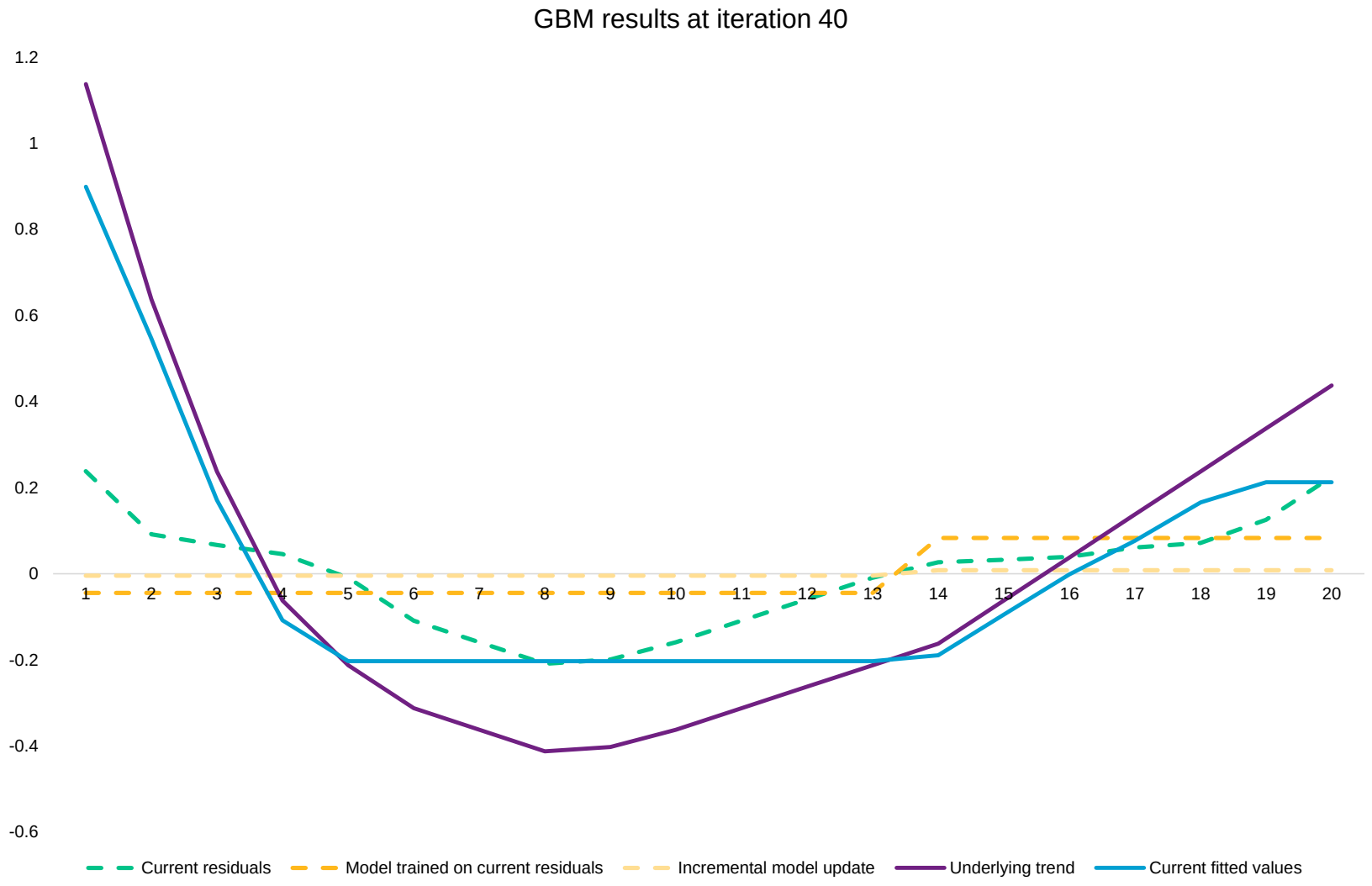
A simple example



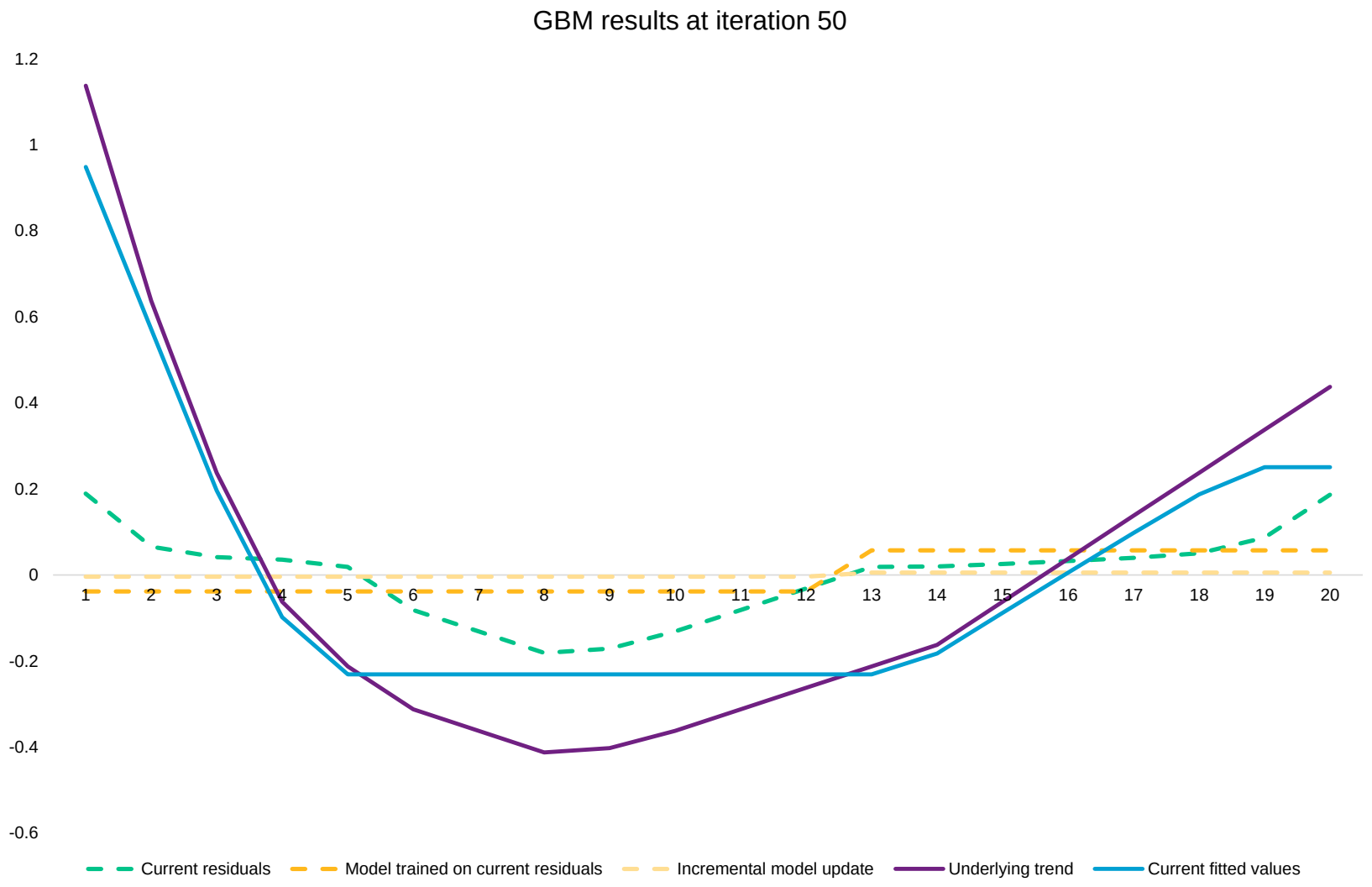
A simple example



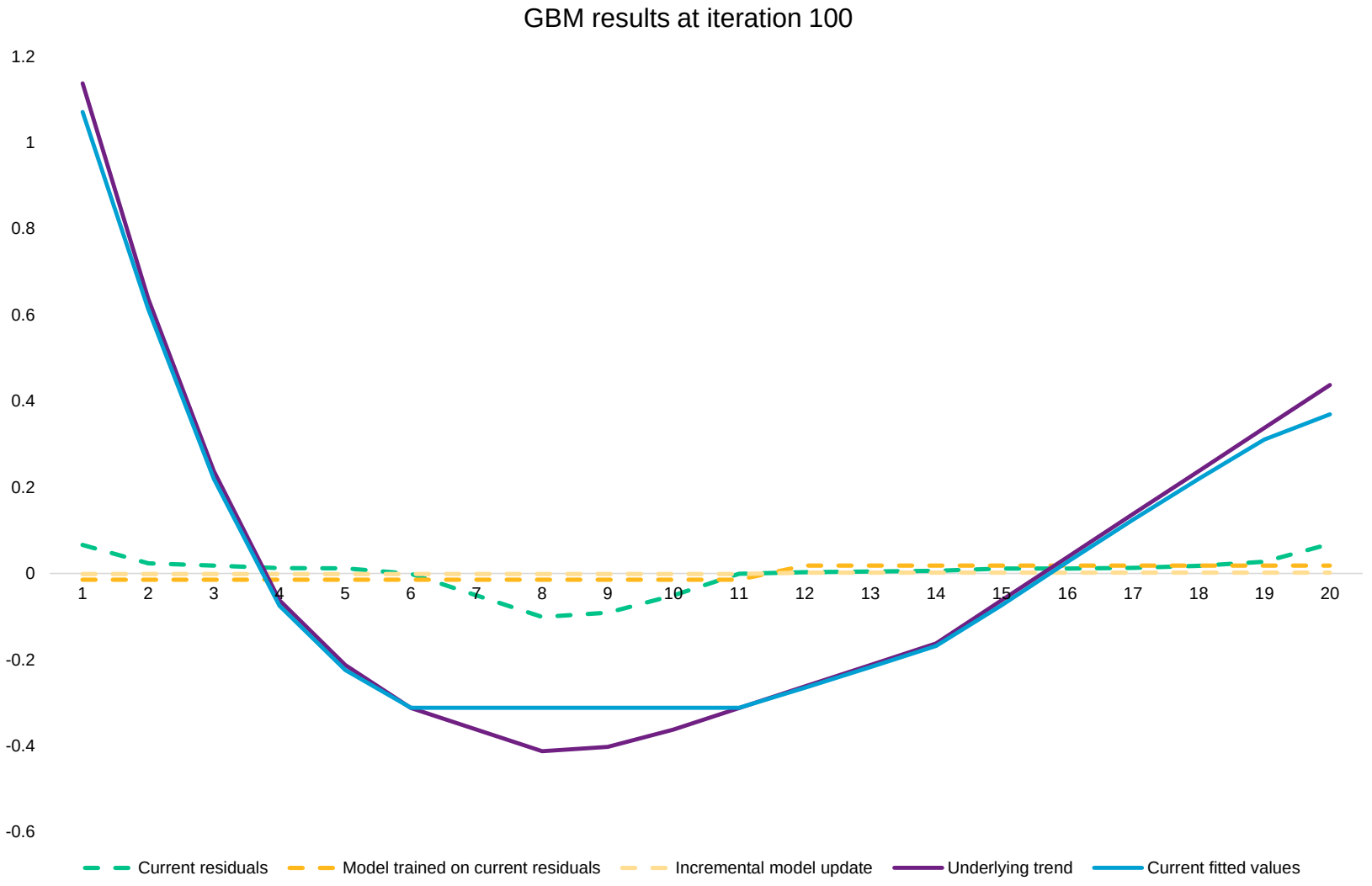
A simple example



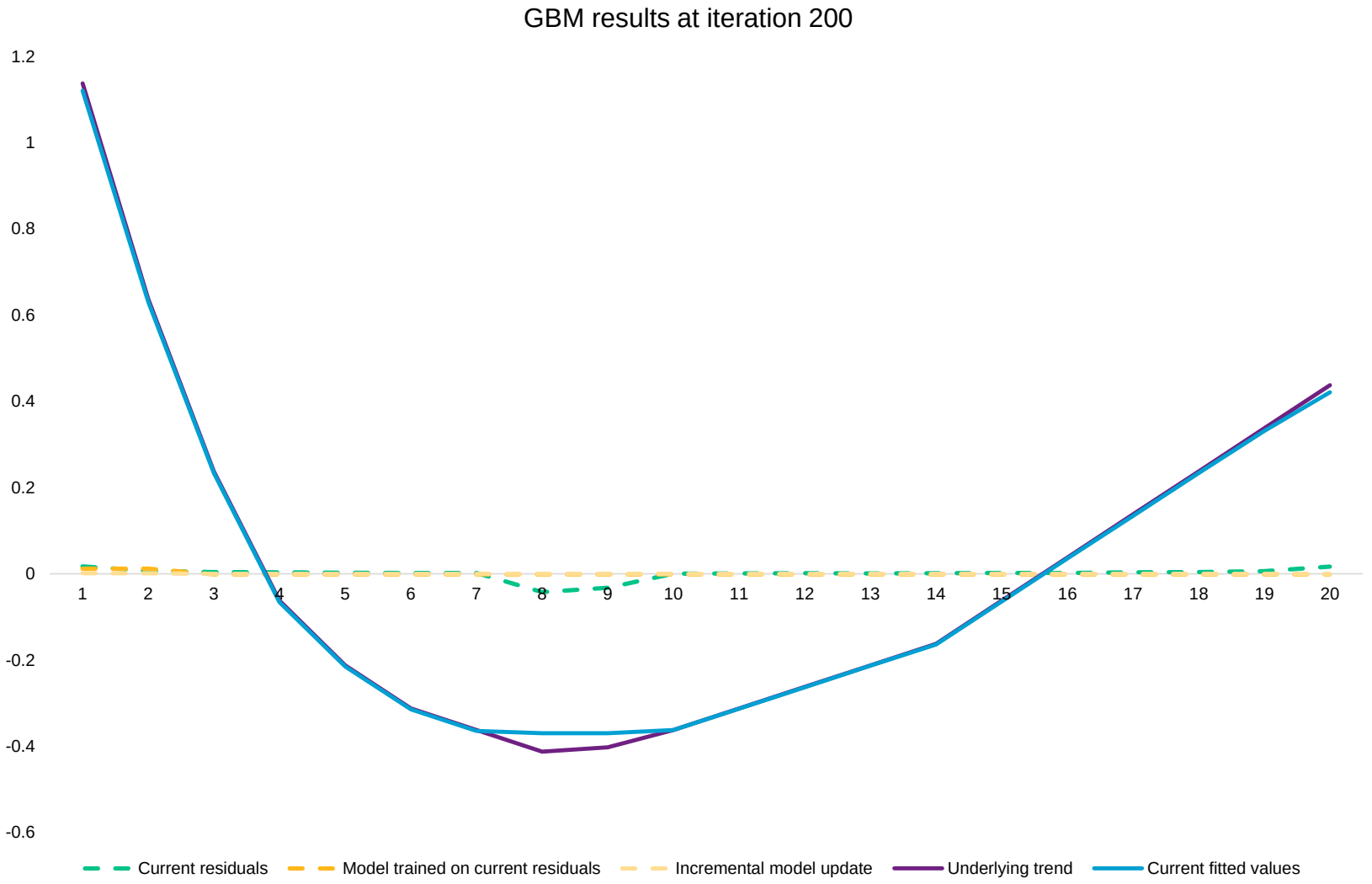
A simple example



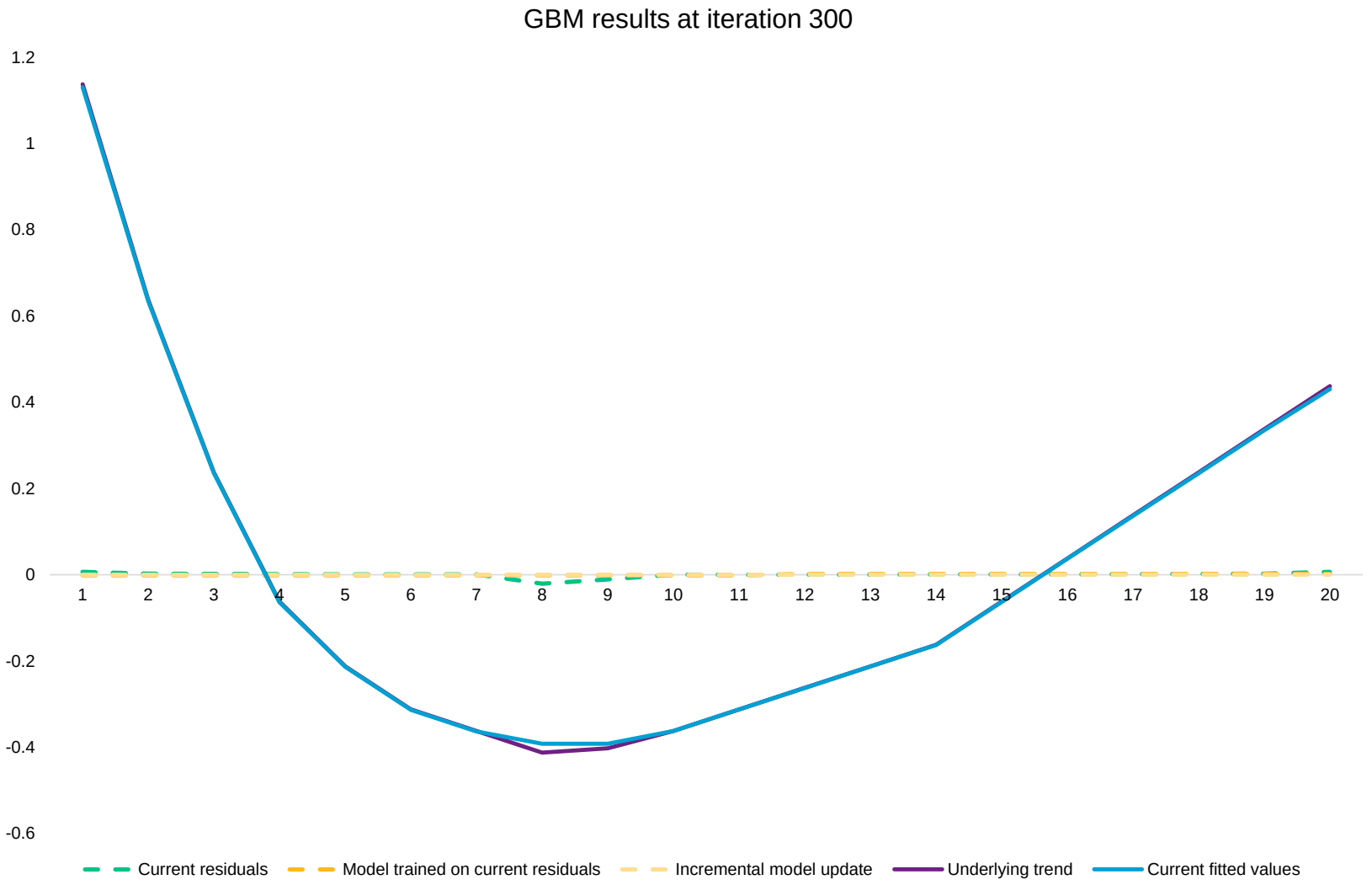
A simple example



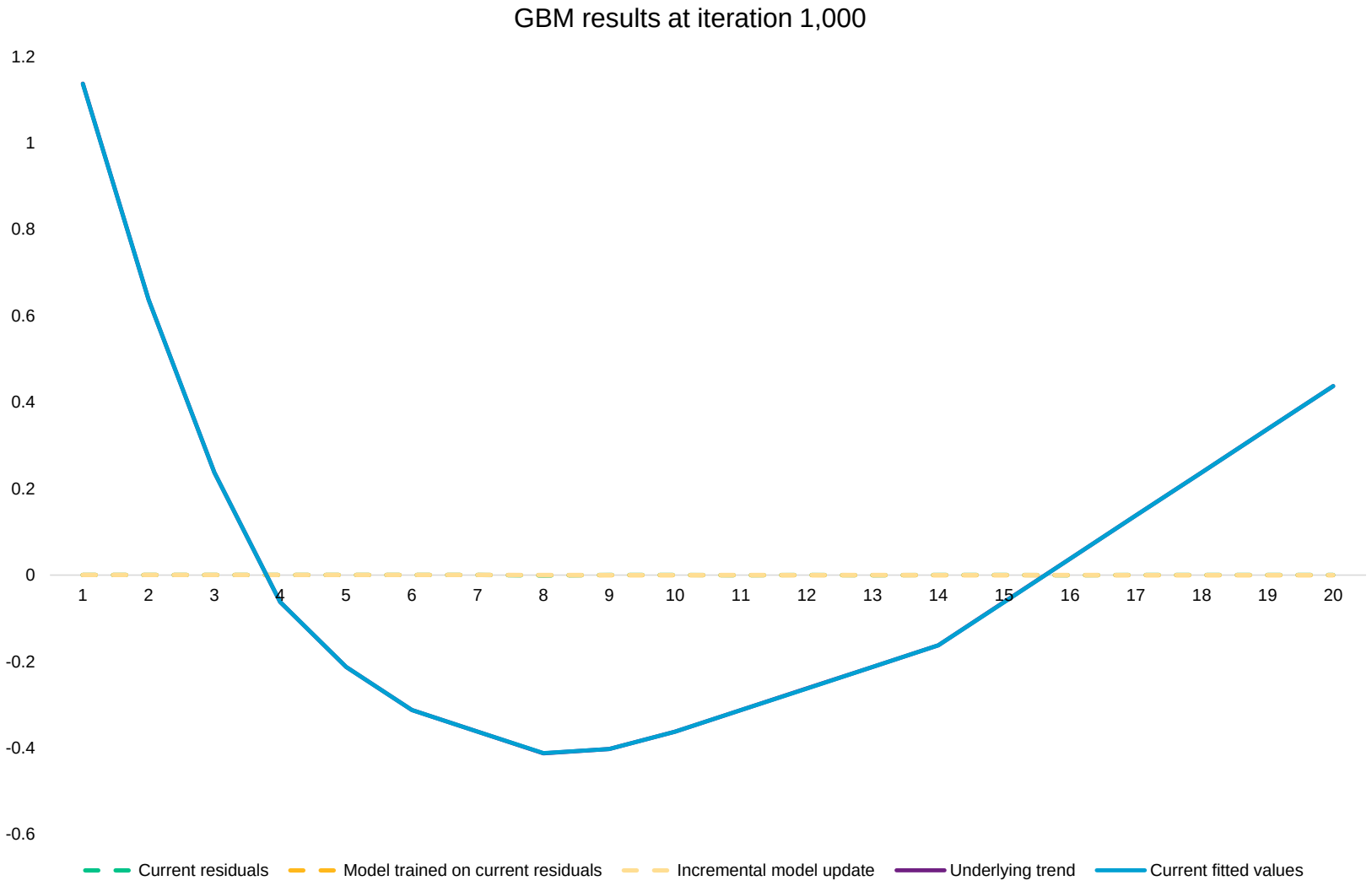
A simple example



A simple example

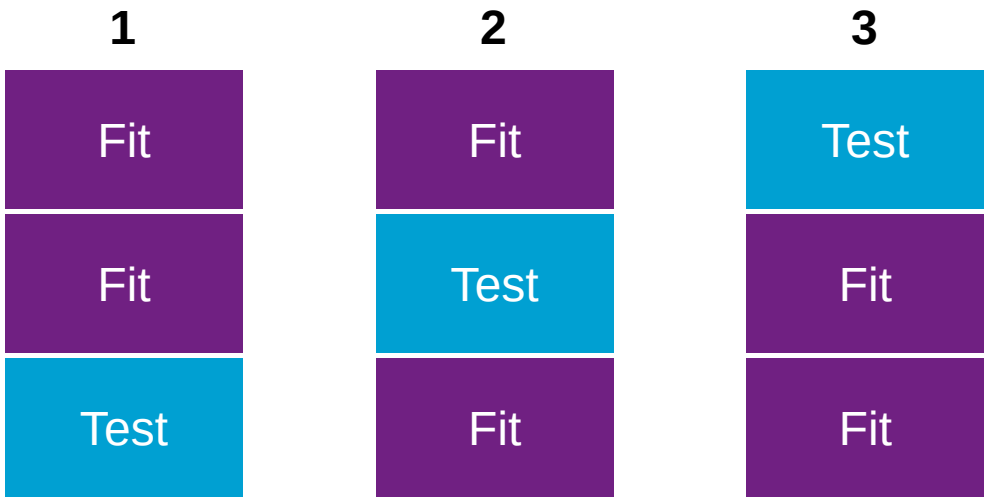


A simple example



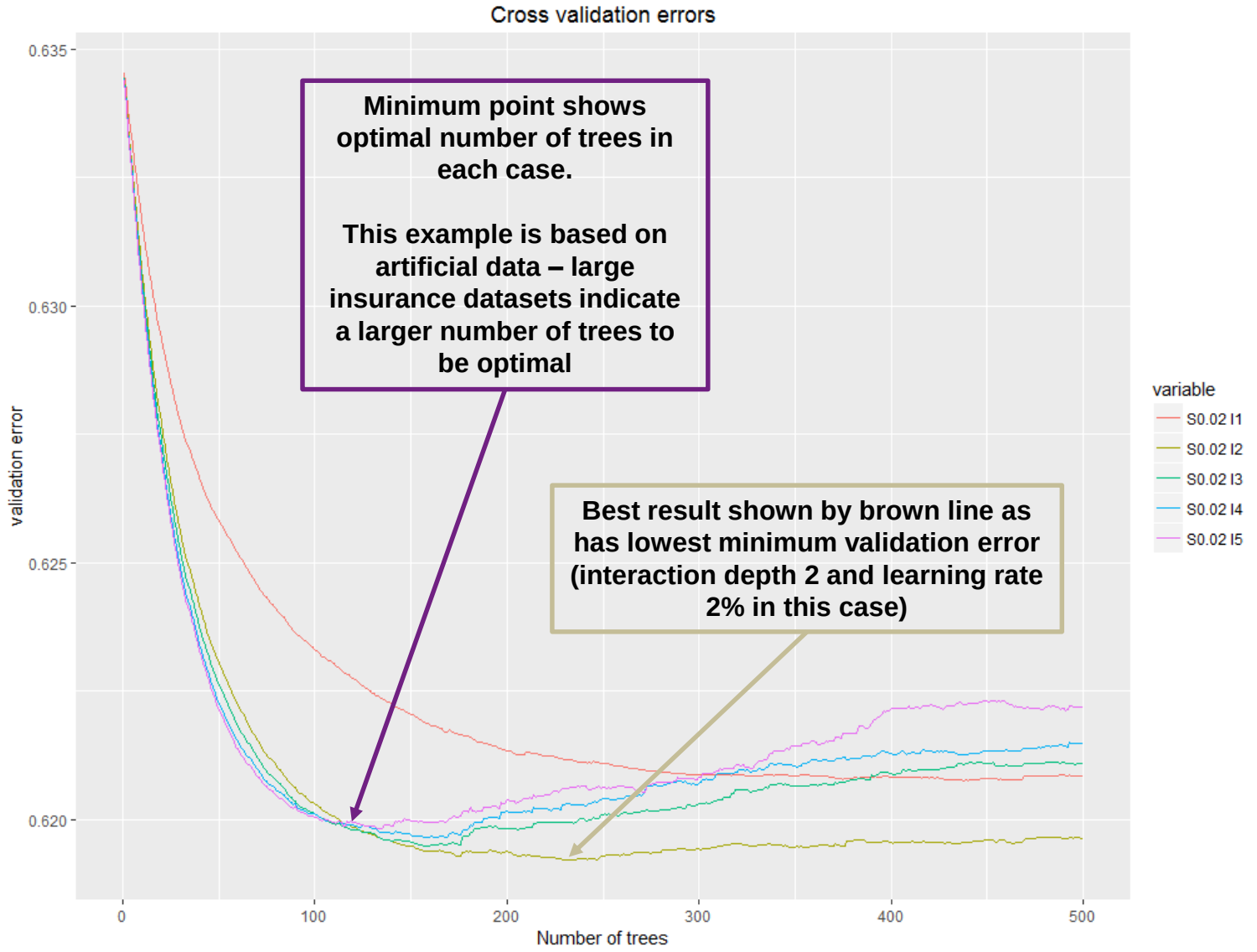
Calibrating the assumptions

- n-fold cross validation used to develop the interaction depth and learning rate assumptions
  - Eg for 3-fold validation, split into 3, fit on purple, test on blue parts, take average

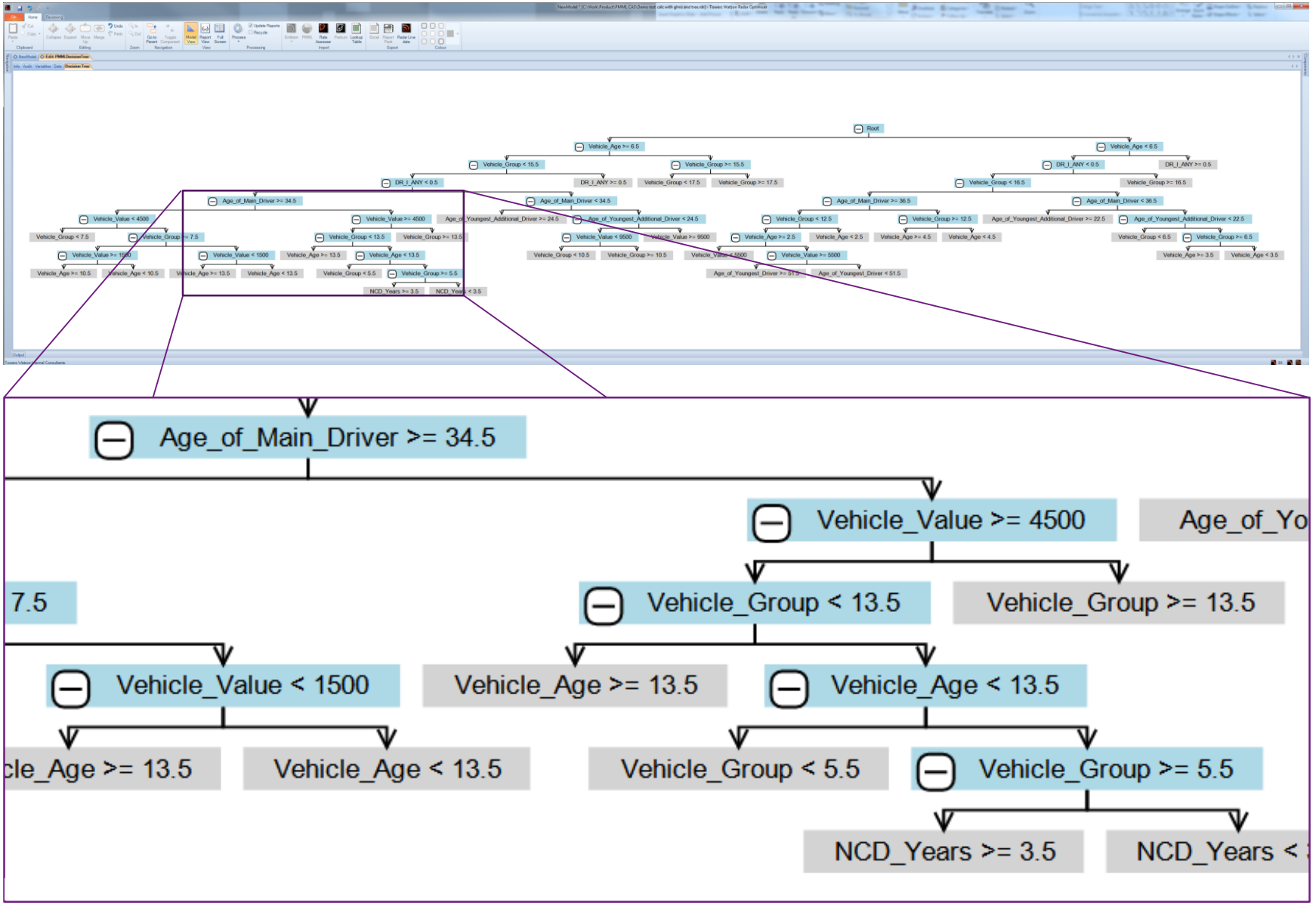


- Resulting plots can be used to determine the optimal assumption choice
  - Including how many trees to run

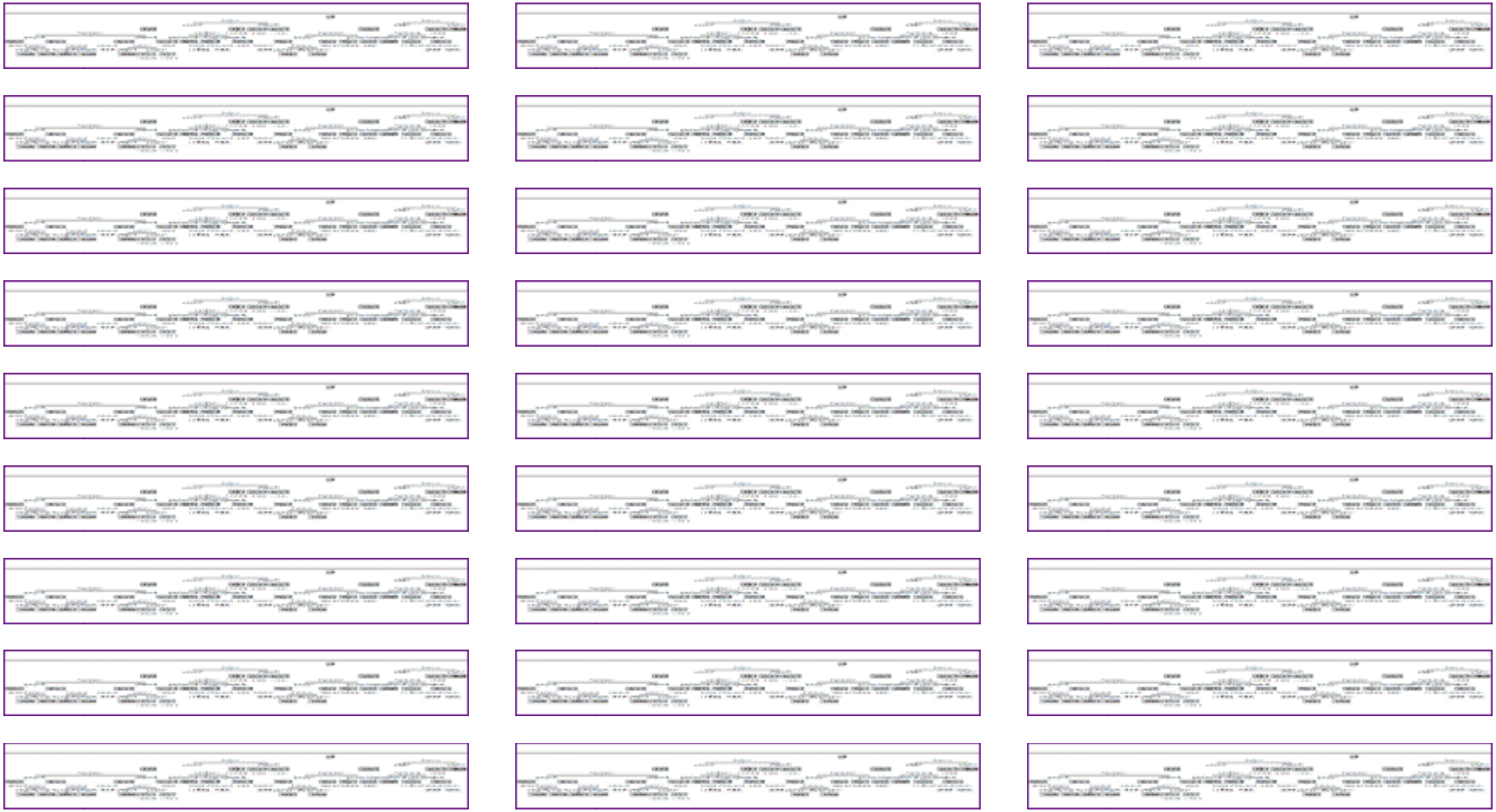
Example 5-fold cross validation

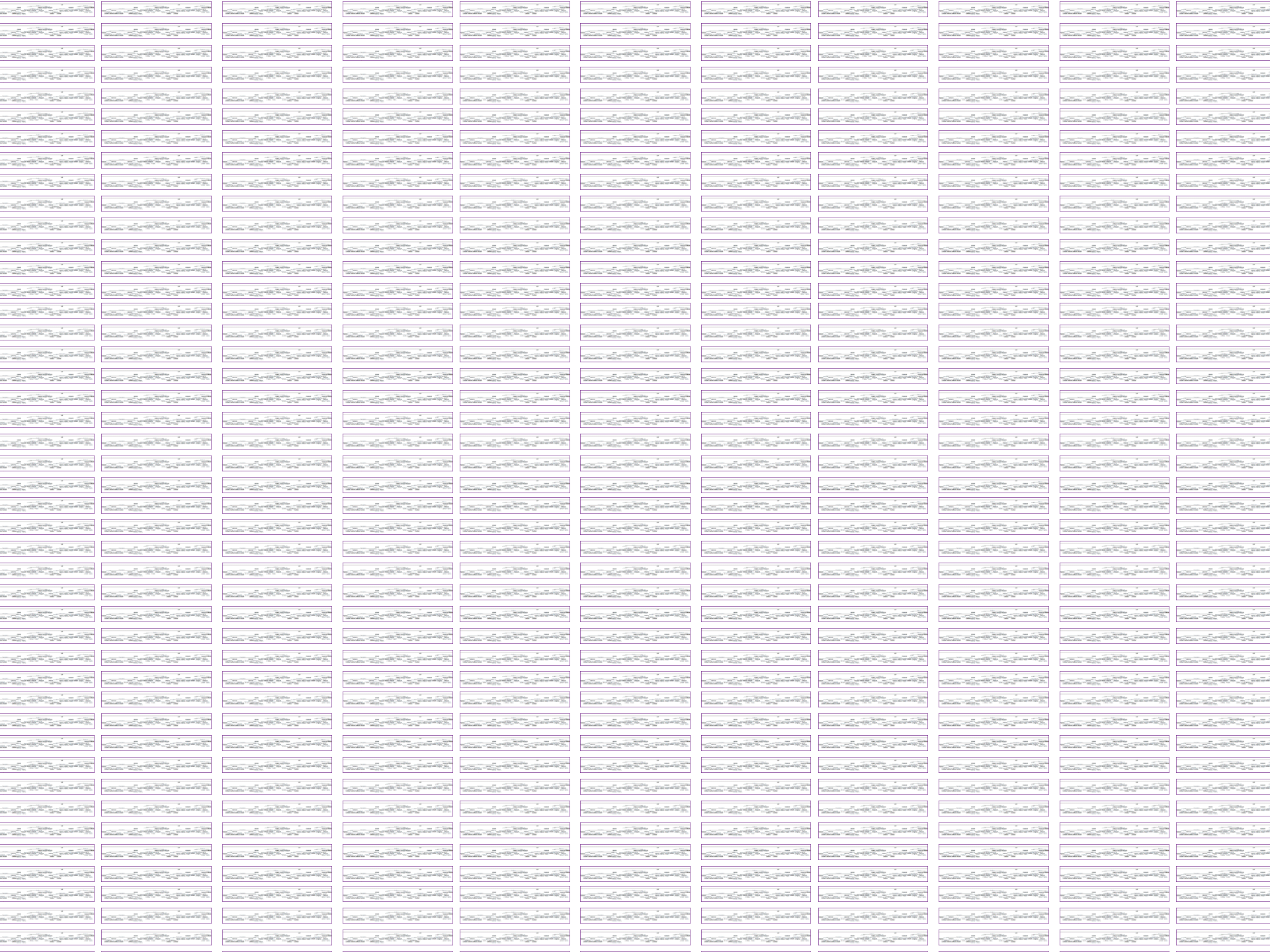


What does the result look like?



What does the result look like?





Three (and a half) interesting questions

1. Does the model add value?
2. What does the model mean?
   - Do we even need to know?
3. How can we use the model?

Case study

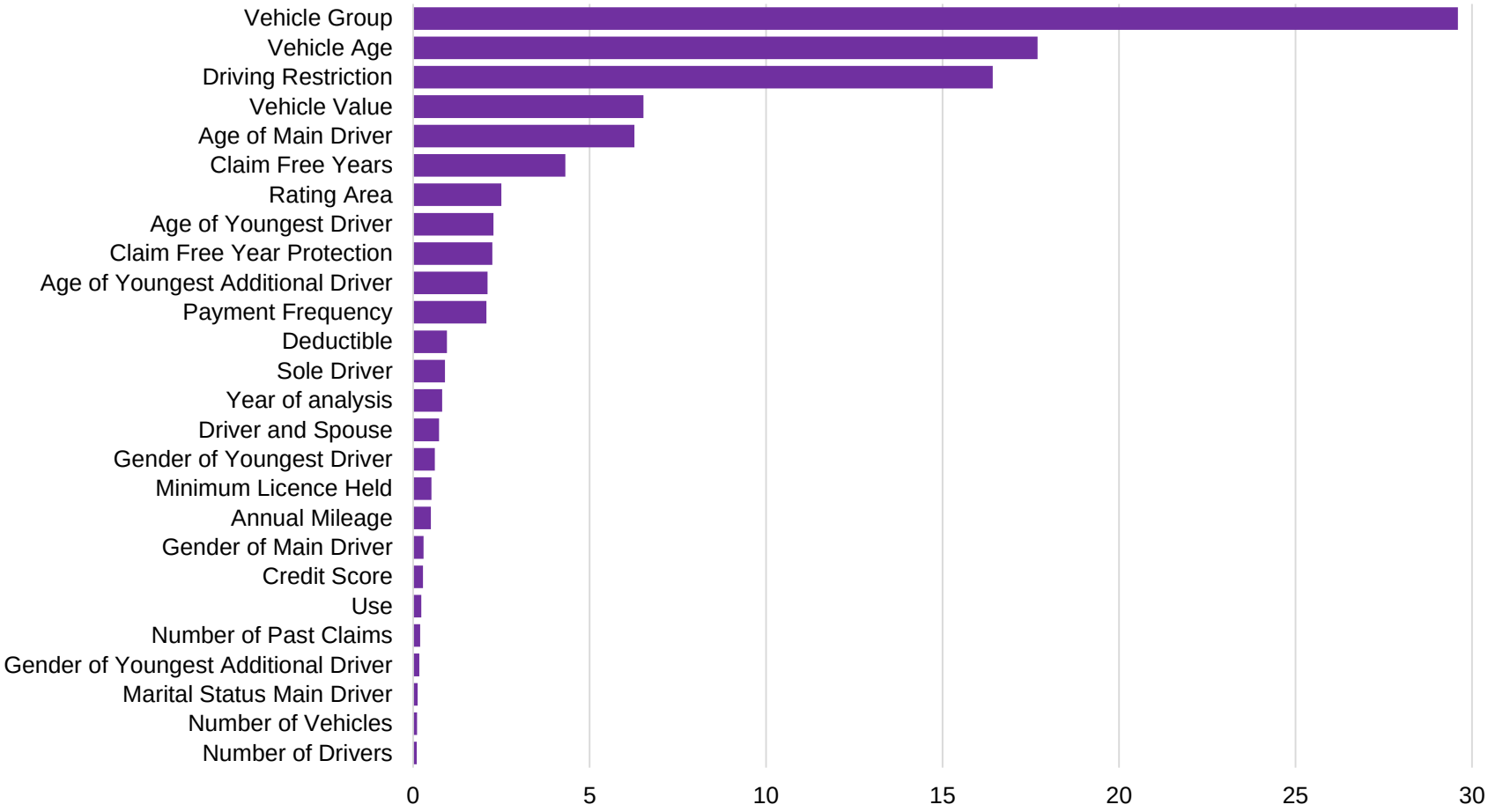
Redacted for posted presentation

Three (and a half) interesting questions

1. Does the model add value?
2. What does the model mean?
   - Do we even need to know?
3. How can we use the model?

Factor importance – relative influence

The relative influence of a factor can be measured as the total reduction in error attributable to splits by that factor, across all trees in the GBM



Partial dependency plots

- View the model as a function of one or two variables after accounting for the average effects of the other variables

$$f_a(X_a) = E_{X_b} f(X_a, X_b) \approx \frac{1}{N}\sum_{i=1}^{N} f(X_a, x_{ib})$$

- This is not the same as ignoring the effects of the other variables

$$\tilde{f}_a(X_a) = E(f(X_a, X_b)|X_a)$$

- For example:

X^1	X^2	X^3	$f(X)=X^1+X^2+X^3$
1	0	11	12
10	2	2	24
2	1	10	13

For each point X^1_i to be plotted:

- Calculate the N predicted values f(X) using X^1_i and each of the j^{th} values of the other variable : $f(X^1_i, X^2_j, X^3_j, \ldots)$ for each j
- Sum these predicted values together and divide by N, effectively averaging out the effects of the other variables
- Repeat for each point X^1_i over the range of the plot

$$f_{X_i^1}(X_i^1) = \frac{1}{3}\sum_{j=1}^{3} f(X_i^1, X_j^2, X_j^3) = \frac{1}{3}(X_i^1 + 11 + X_i^1 + 4 + X_i^1 + 11) = X_i^1 + \frac{26}{3}$$

$$f_{\{X_i^1, X_j^2\}}(X_i^1, X_j^2) = \frac{1}{3}\sum_{k=1}^{3} f(X_i^1, X_j^2, X_k^3) = X_i^1 + X_j^2 + \frac{23}{3}$$

Partial dependency plots

X^1	X^2	X^3	$f(X)=X^1+X^2+X^3$
1	0	11	12
10	2	2	24
2	1	10	13

$X^1=1$

X^1	X^2	X^3	$f(X)=X^1+X^2+X^3$
1	0	11	12
1	2	2	5
1	1	10	12

9.67

$X^1=10$

X^1	X^2	X^3	$f(X)=X^1+X^2+X^3$
10	0	11	21
10	2	2	14
10	1	10	21

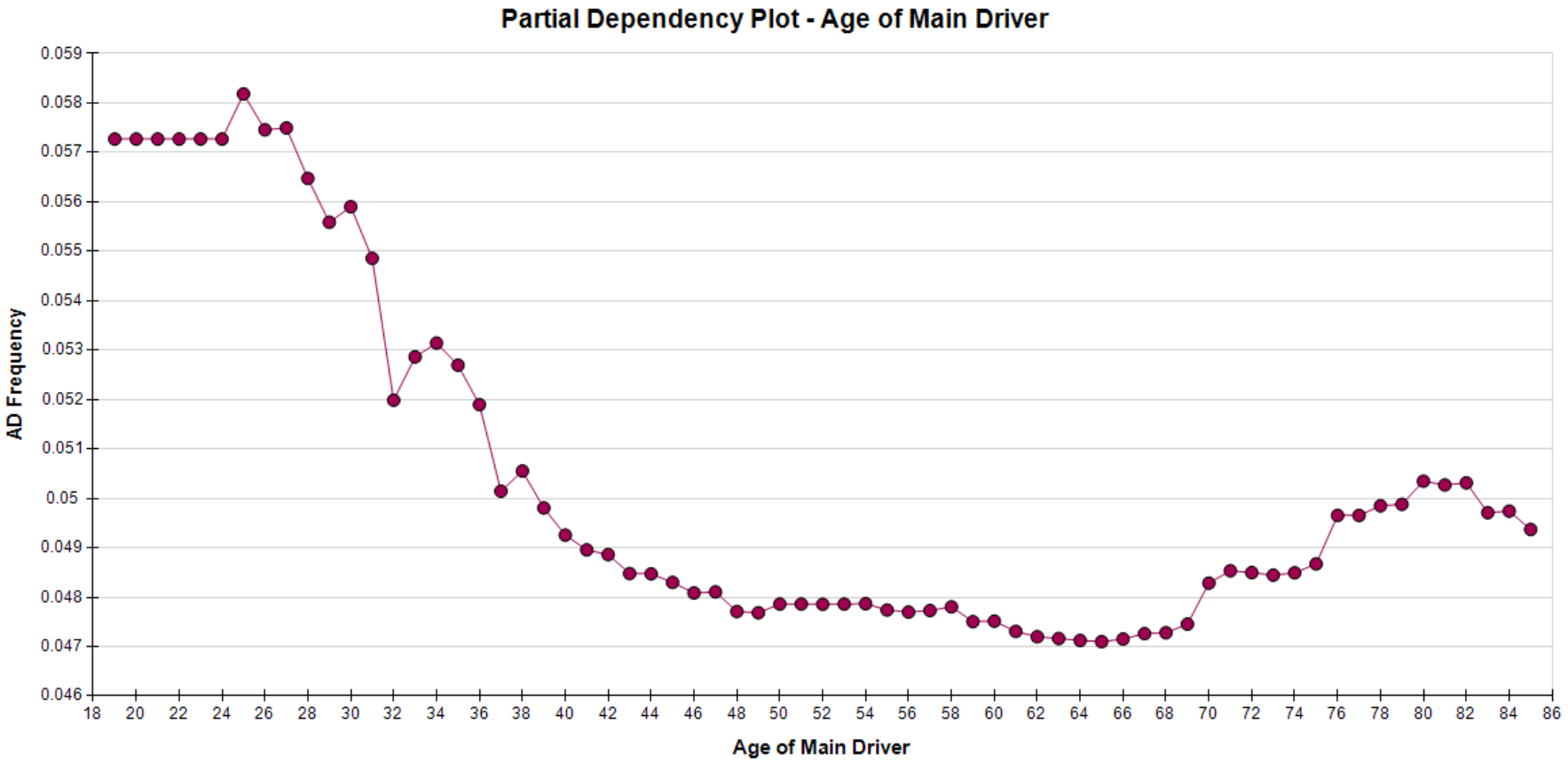
18.67

$X^1=2$

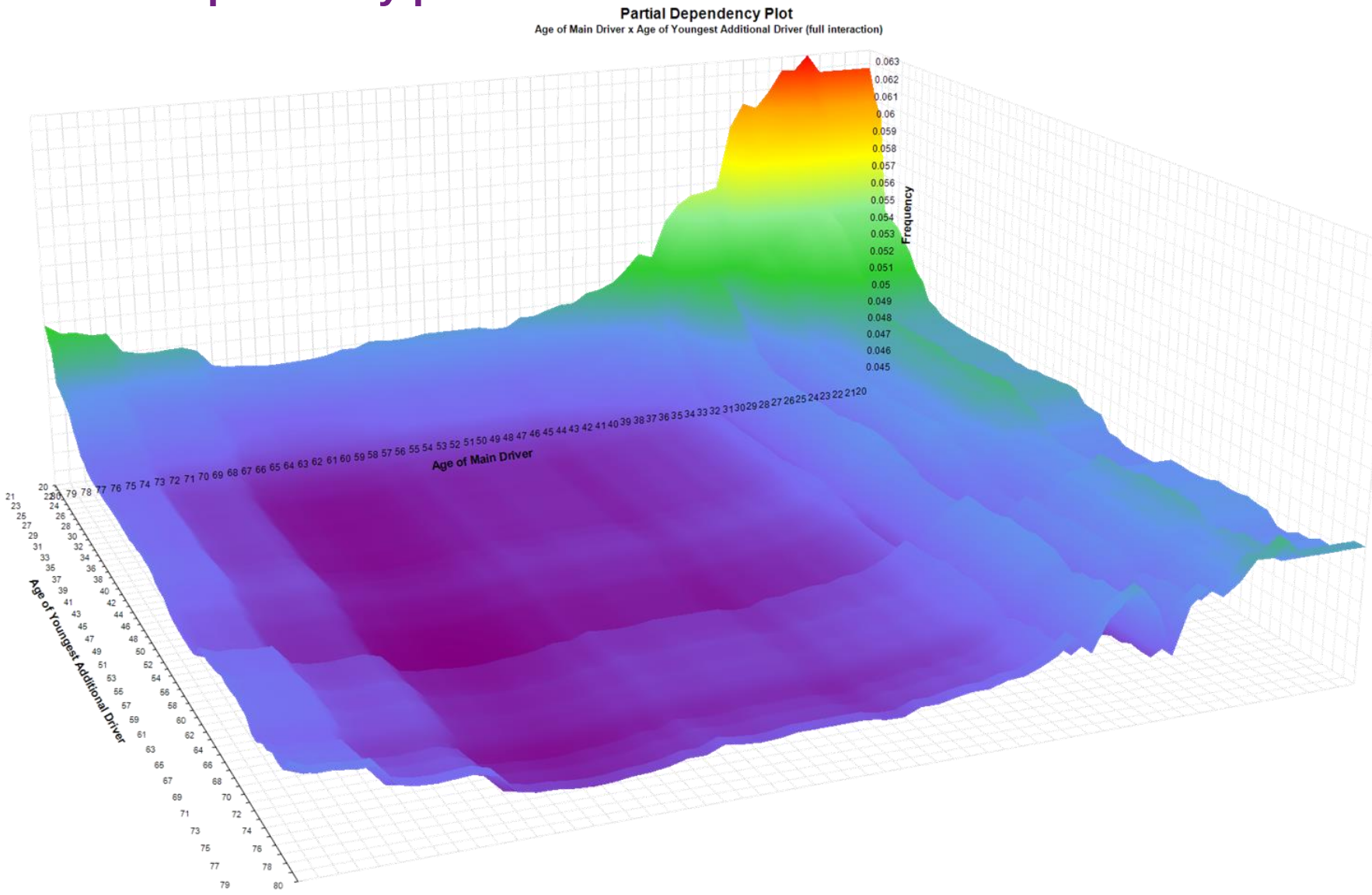
X^1	X^2	X^3	$f(X)=X^1+X^2+X^3$
2	0	11	13
2	2	2	6
2	1	10	13

10.67

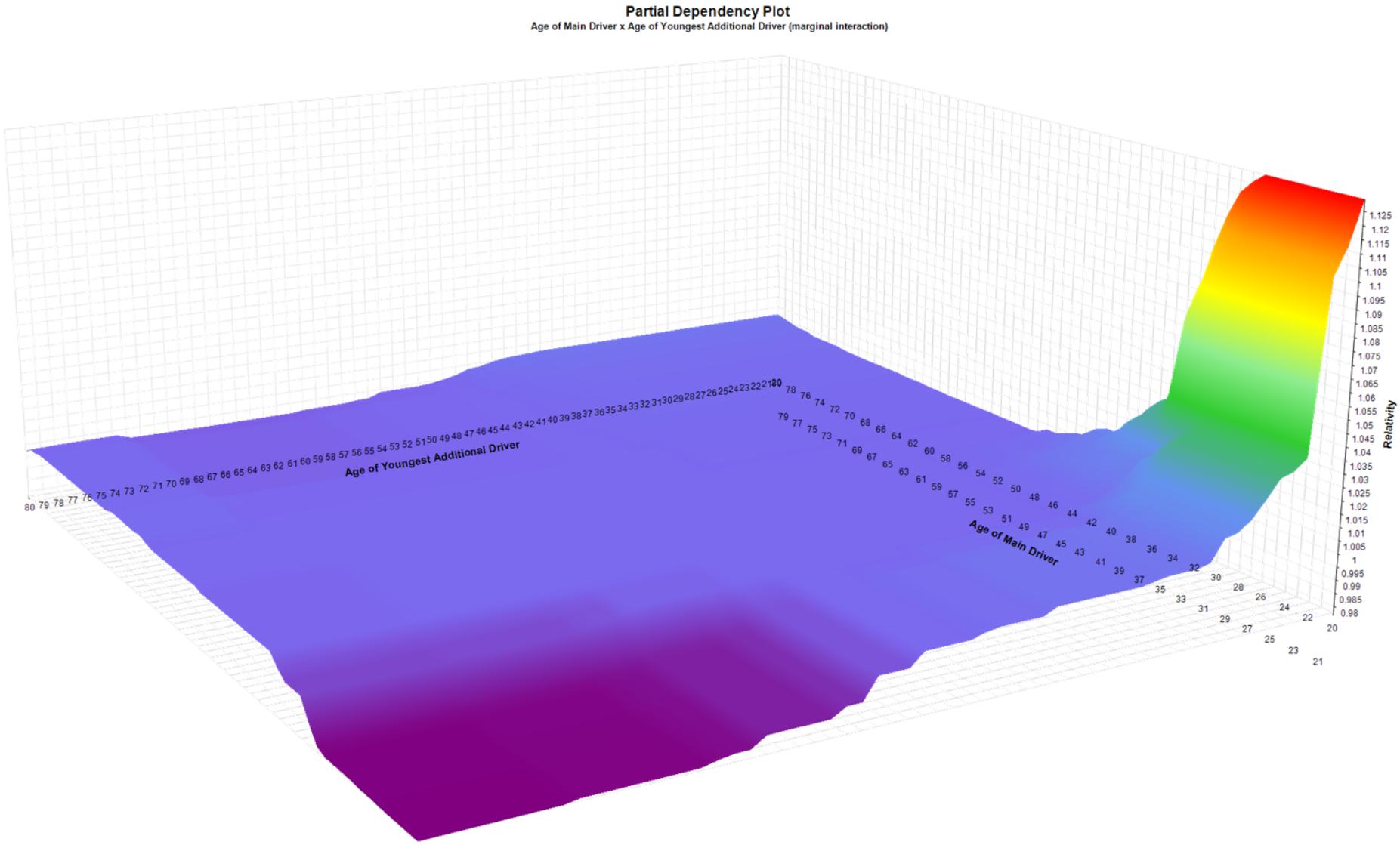
Partial dependency plots



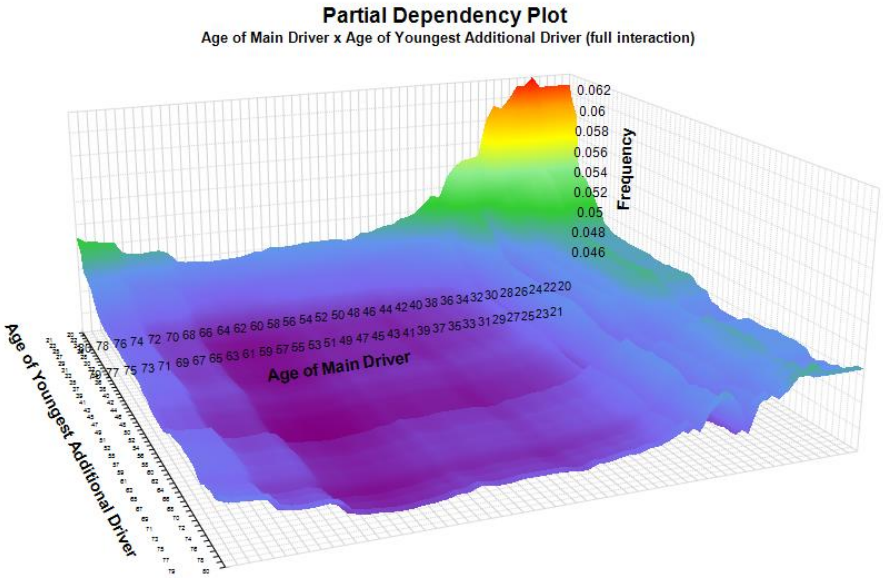
Partial dependency plots

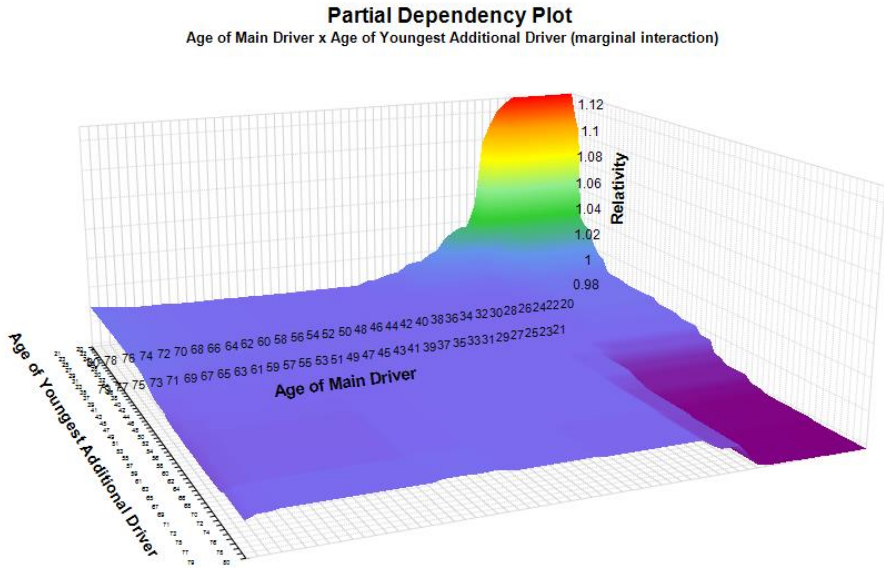


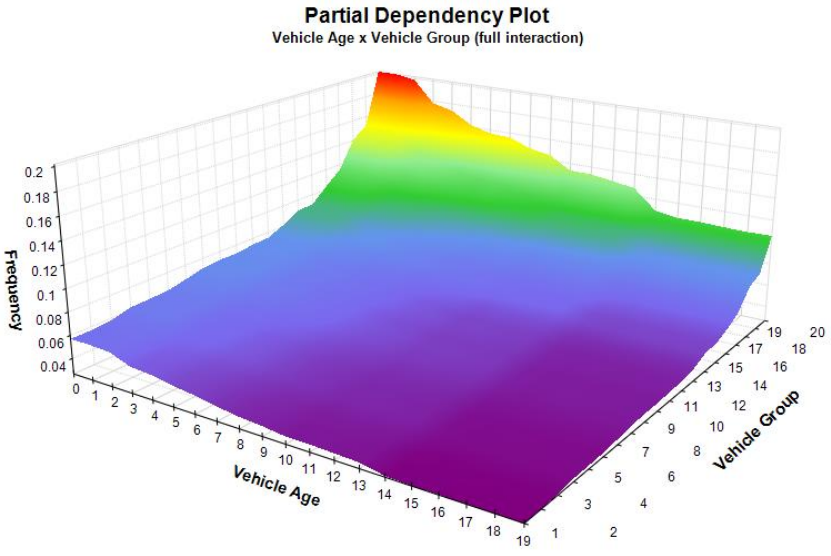
Partial dependency plots

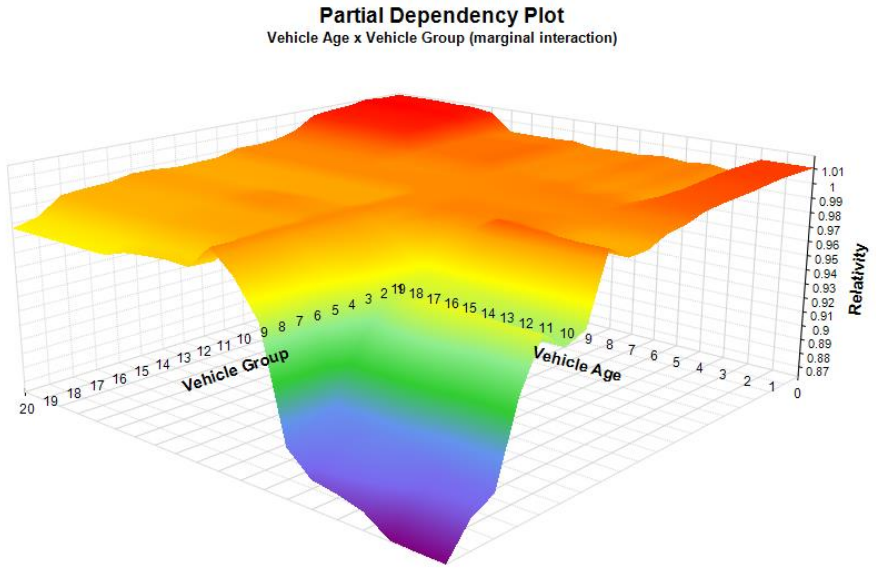


Partial dependency plots

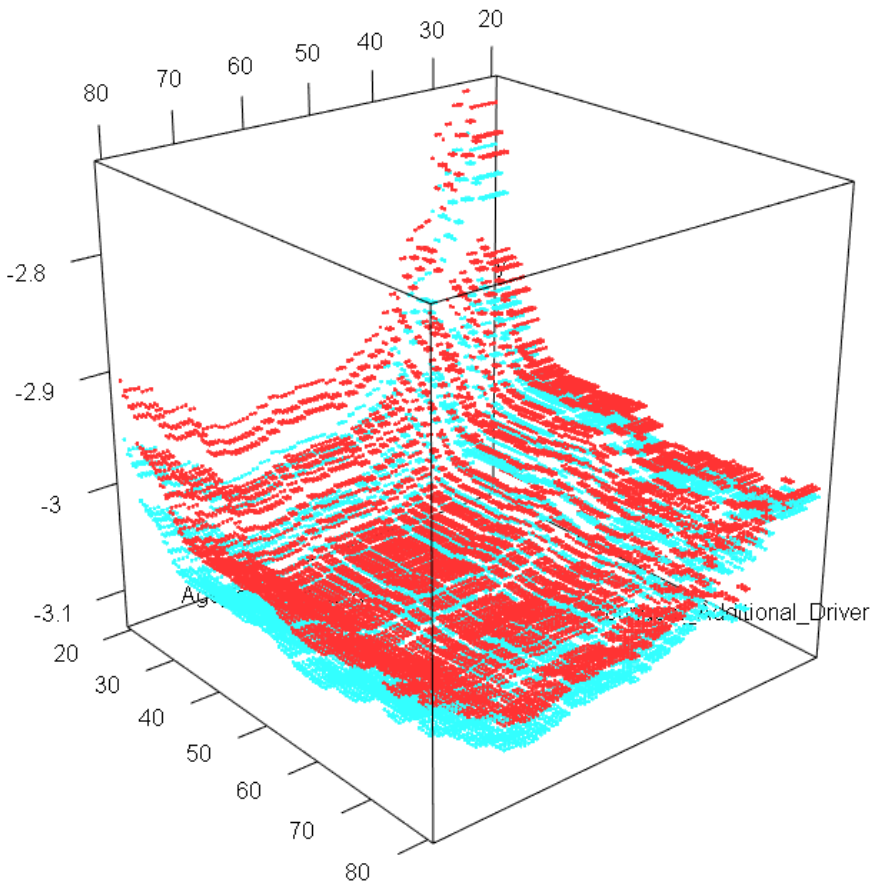








Partial dependency plots



Advantages

- Qualitative description of properties of relationships
- Most revealing of additive and multiplicative relationships

Disadvantages

- "GLM view of a non-GLM thing"
- Interaction effects outside of the chosen subset may be obfuscated
- eg if X_1X_2 is important and X_2 is averaged out in the partial dependence plot, X_1 may show as being heterogeneous, thus obfuscating the complexity of the modelled relationships

Three (and a half) interesting questions

1. Does the model add value?
2. What does the model mean?
   - Do we even need to know?
3. How can we use the model?

How can we use the model?

A. Model down into a traditional table form
B. Use insights to guide traditional GLM
C. Use non-GLM directly

How can we use the model?

A. Model down into a traditional table form
B. Use insights to guide traditional GLM
C. Use non-GLM directly

Case study

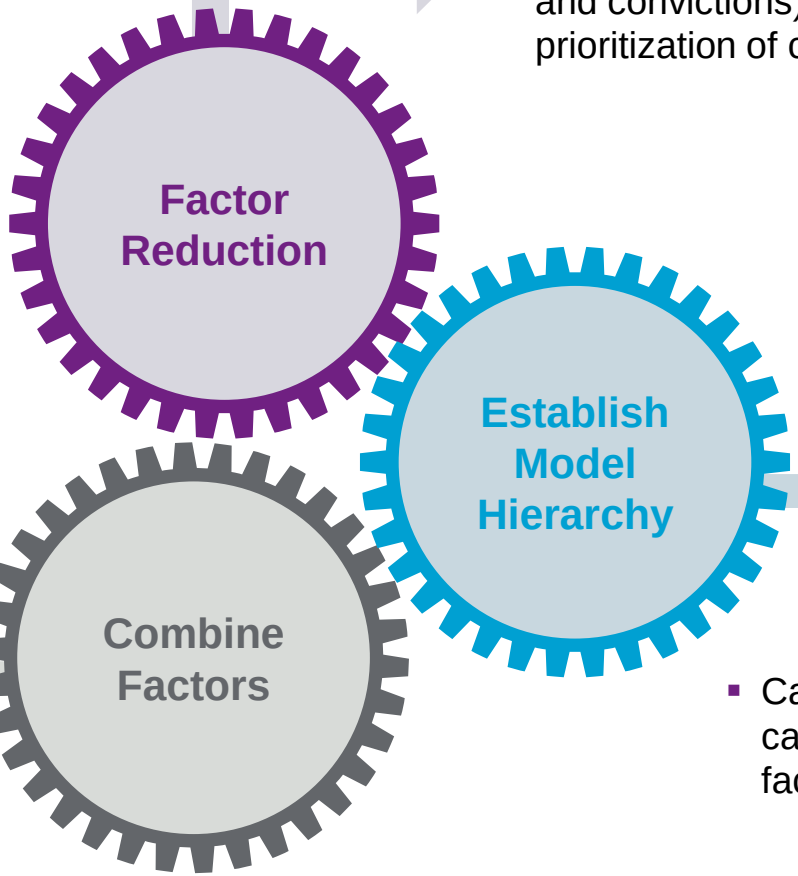
Redacted for posted presentation

How can we use the model?

A. Model down into a traditional table form
B. Use insights to guide traditional GLM
C. Use non-GLM directly

Use insights to guide traditional GLM

- Given many hundreds of factors describing one general characteristic (eg past claims and convictions) GBM can help with prioritization of candidate factors



- Findings of GBM can be restated as higher order interactions within a GLM framework
- For example to correct “corners” of a GLM where the model over predicts

- Can support identification of candidate model segmentation factors within a GLM framework

How can we use the model?

A. Model down into a traditional table form
B. Use insights to guide traditional GLM
C. Use non-GLM directly

How can we use the model?

A. Model down into a traditional table form
B. Use insights to guide traditional GLM
C. Use non-GLM directly

Regulatory issues

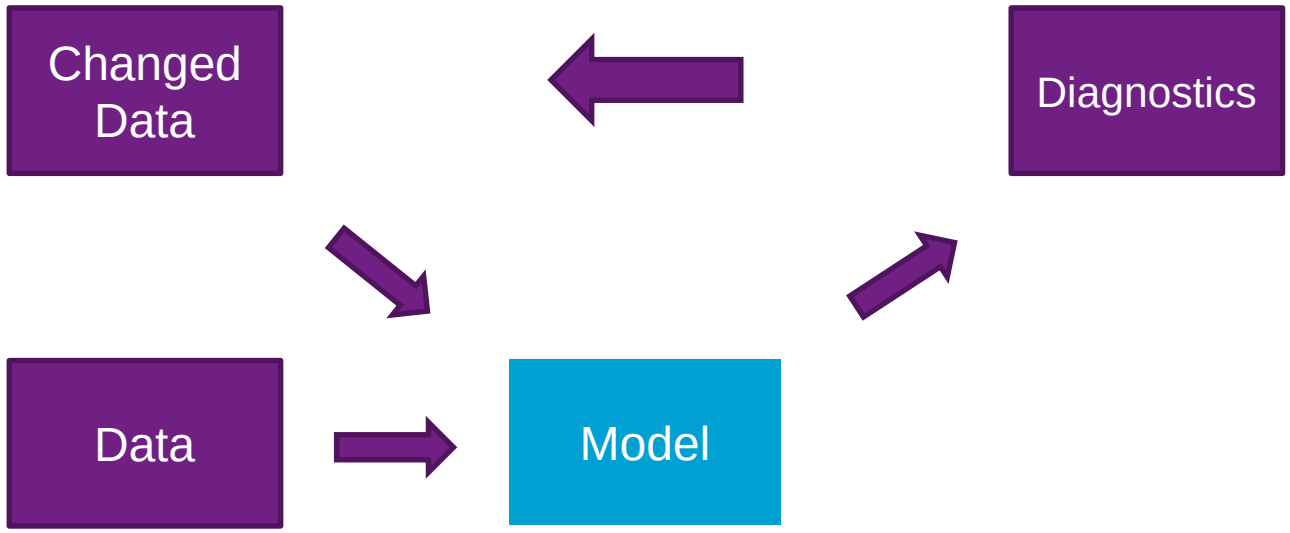
- There are market precedents of machine learning techniques being filed in the US
  - Mostly involving variable/tier creation
- New rating variables (or tiers) created from machine learning techniques:
  - May require some level of interpretability by regulator
  - May raise concerns about nature of data used or size of individual classes and related credibility
- Rate regulation will generally require a closed form solution for rating

Public policy issues

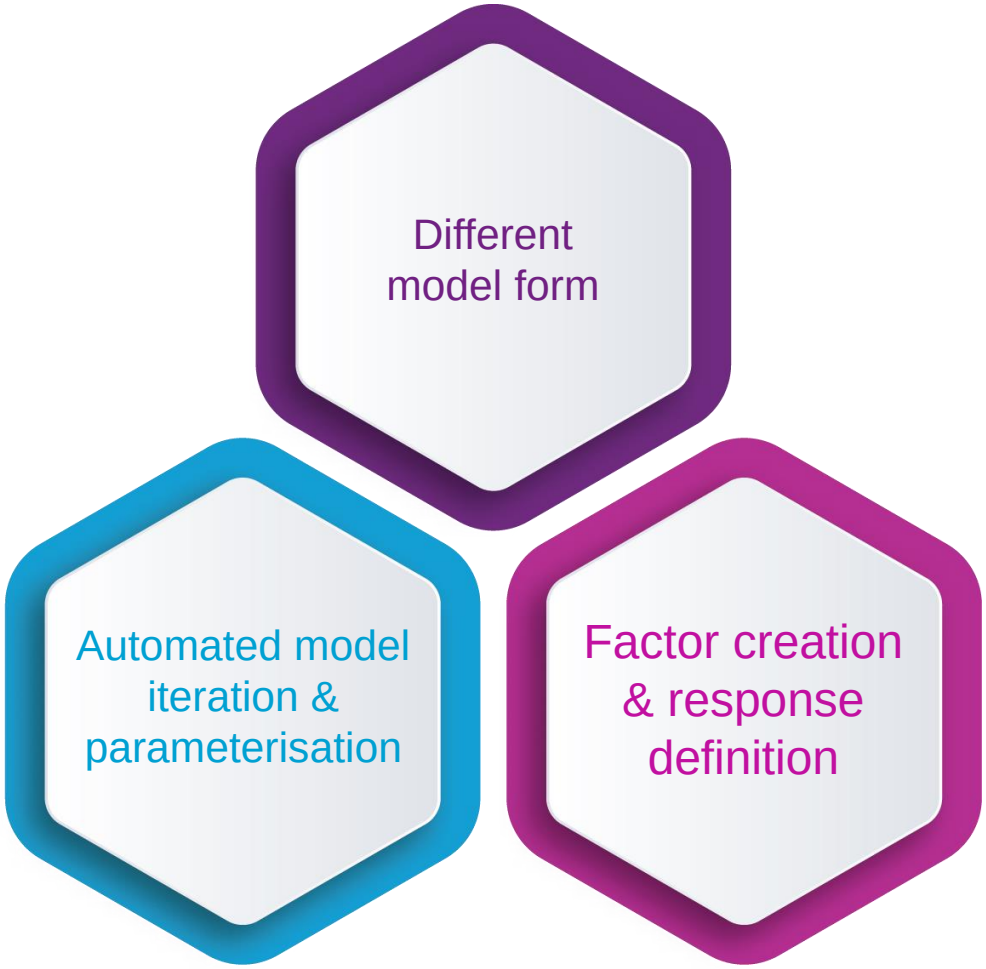
- As rating plans become more granular there may be heightened scrutiny around disparate impact (rating bias for a protected class)
- Complex models like GBM may inadvertently load small segments inappropriately

How can we use the model?

A. Model down into a traditional table form
B. Use insights to guide traditional GLM
C. Use non-GLM directly

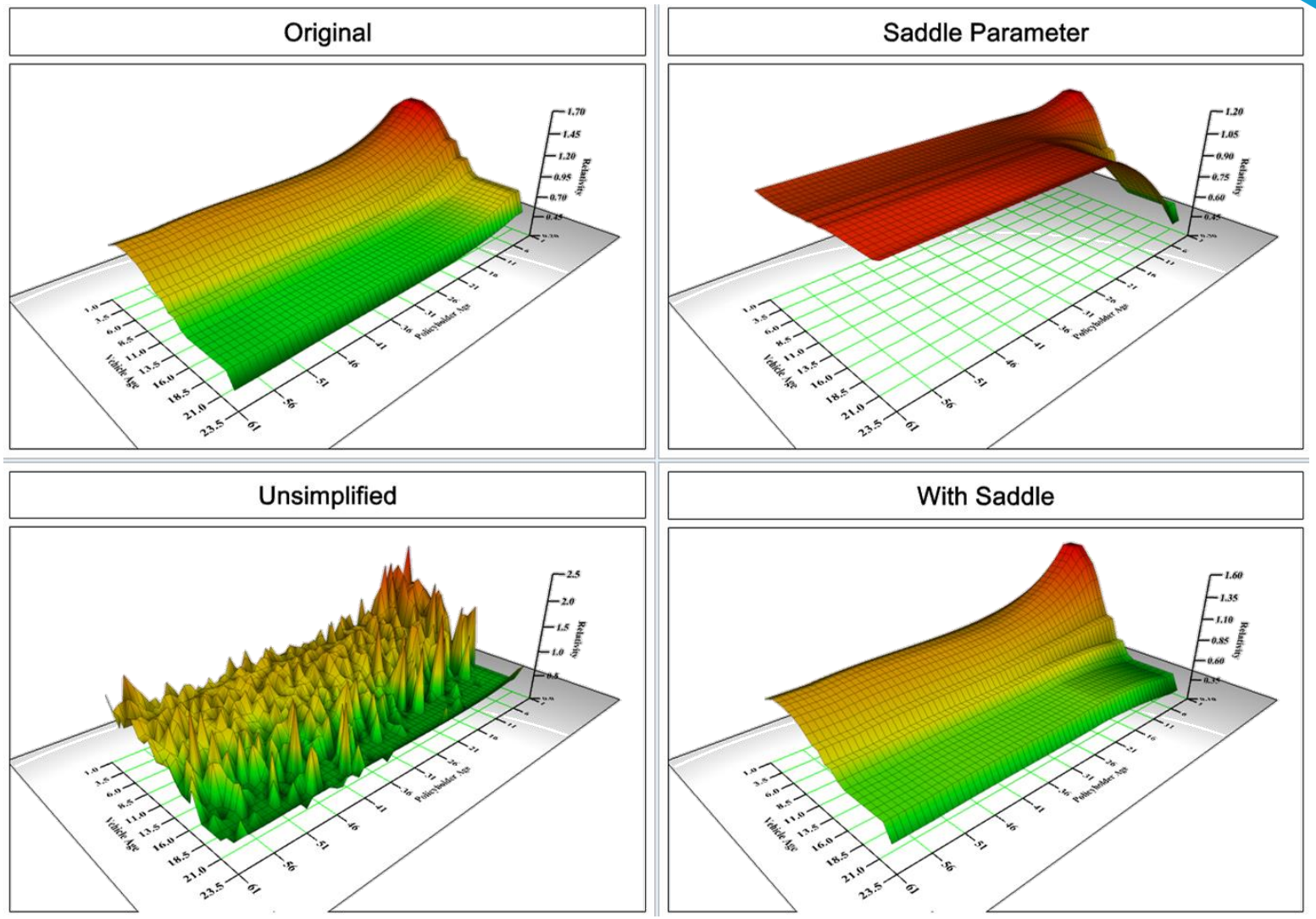


What's really going on here?



Automated parameterization in a GLM

Automated model iteration & parameterisation

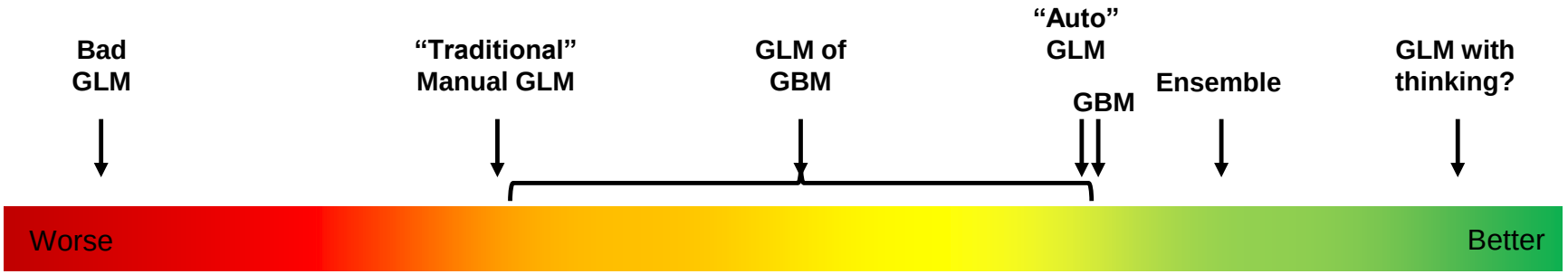


Case study

Redacted for posted presentation

Conclusions

Conclusions



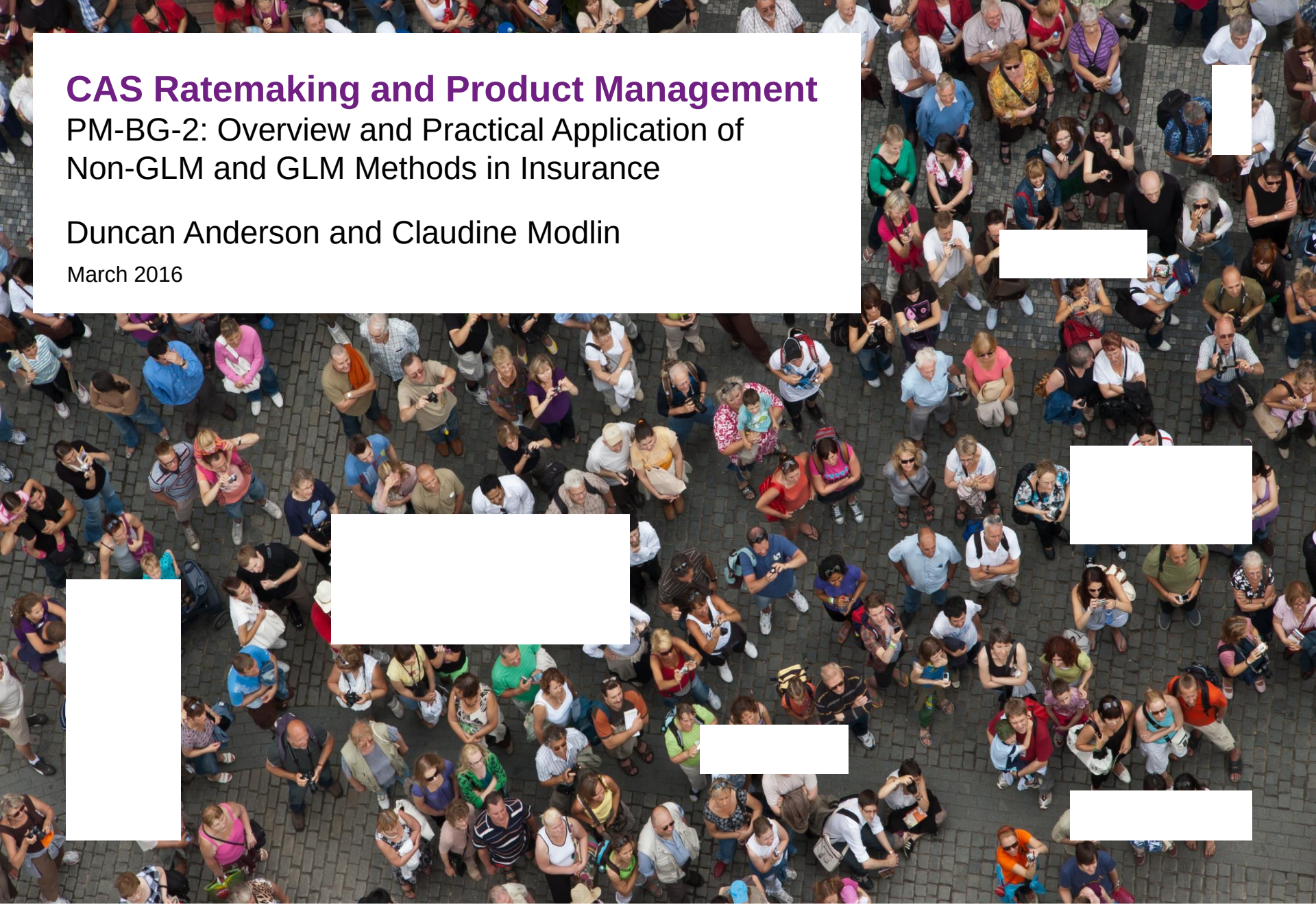
- If you can..
  - Cope with not seeing the model and instead using broad diagnostics
  - And cope with small segments being wrong
  - And your regulator can as well
  - And you have a rating engine that can implement it
  - And you have the software and hardware to fit to large datasets
- …then there are some predictive lift benefits from GBMs et al in pricing
  - In other areas, eg marketing, application is less problematic
- If not, there are ways of finding new insight, implementing within GLMs
- But also if you accept models that are hard to interpret, GLMs can be machine fitted also…
- Perhaps most important don't lose sight of the value of thinking and domain expertise

CAS Ratemaking and Product Management

PM-BG-2: Overview and Practical Application of Non-GLM and GLM Methods in Insurance

Duncan Anderson and Claudine Modlin

March 2016



Willis Towers Watson